DOCUMENT RESUME

ED 296 701 IR 013 325

AUTHOR	Dede, Christopher J.; And Others
TITLE	Intelligent Computer-Assisted Instruction: A Review and Assessment of ICAI Research and Its Potential for Education.
INSTITUTION	Educational Technology Center, Cambridge, MA.
SPONS AGENCY	National Inst. of Education (ED), Washington, DC.
PUB DATE	May 85
CONTRACT	400-83-0041
NOTE	117p.
PUB TYPE	Information Analyses (070) -- Reports - Descriptive (141)
EDRS PRICE	MF01/PC05 Plus Postage.
DESCRIPTORS	*Artificial Intelligence; *Computer Assisted Instruction; *Computer Software; Computer System Design; Mathematical Models; State of the Art Reviews; Technological Advancement
IDENTIFIERS	*Intelligent CAI Systems; Knowledge Representation

ABSTRACT

The first of five sections in this report places intelligent computer-assisted instruction (ICAI) in its historical context through discussions of traditional computer-assisted instruction (CAI) linear and branching programs; TICCIT and PLATO IV, two CAI demonstration projects funded by the National Science Foundation; generative programs, the earliest application of artificial intelligence in education; mathematical models of learning; and recent developments such as dialogue-based tutorial systems. The second section describes ICAI systems and their components, i.e., the expertise module and types of knowledge representation schemes; the student module; and the communication module. Several examples of ICAI systems--SCHOLAR, WHY, SOPHIE, BUGGY & DEBUGGY, GUIDON, WEST, and GEOMETRY TUTOR--are presented in the third section, and the fourth looks at major themes in ICAI research and the current capabilities of ICAI systems. The fifth section discusses the potentials of ICAI for education and its implications for teachers, students, administrators, and researcher/developers. (77 references) (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED296701

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

[X] This document has been reproduced as received from the person or organization originating it.

[ ] Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

INTELLIGENT COMPUTER-ASSISTED INSTRUCTION: A REVIEW AND ASSESSMENT OF ICAI RESEARCH AND ITS POTENTIAL FOR EDUCATION

Prepared for

EDUCATIONAL TECHNOLOGY CENTER

by

Christopher J. Dede
Philip P. Zodhiates
Charles L. Thompson

EDUCATION DEVELOPMENT CENTER
55 Chapel Street
Newton, Massachusetts 02160

MAY 1985

This report was prepared with funding from the National Institute of Education (Contract #NIE 400-83-0041). The views expressed herein are those of the authors and do not necessarily reflect the positions or policies of the Institute.

IR013325

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Beth Wilson

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

TABLE OF CONTENTS

Section I: Historical Context 1

Introduction 1
Historical Context 3

Section II: Description of ICAI Systems and Their Components 14

Expertise Module 14
Student Module 22
Teaching Module (to follow)
Communication Module 26

Section III: Examples of ICAI Systems 32

SCHOLAR 32
WHY 35
SOPHIE 38
BUGGY & DEBUGGY 40
GUIDON 43
WEST 47
GEOMETRY TUTOR 51

Section IV: Assessing Intelligent Computer-Assisted Instruction 55

Major Themes in Current Research 55
Expertise Module 55
Student Model 65
Teaching Module 74
Communications Module 79

Capabilities and Limitations of Current ICAI 82

Section V: The Potential of Intelligent Computer-Assisted Instruction for Education 88

Implications for Teachers 95
Implications for Learners 101
Implications for Administrators 103
Implications for Researchers and Developers 105

SECTION I: HISTORICAL CONTEXT

Introduction

The field of artificial intelligence may, at first glance, appear remote and even irrelevant to the practical concerns of school teachers and administrators. At a time when school budgets are already stretched to the limit and the great majority of teachers are only now beginning to assimilate the relatively simple technology of 6502-based microcomputers into the conventional curriculum, it may seem preposterous to suggest that AI, much of it still in the early stages of basic research, and notorious for its need of expensive and powerful machines, has anything of practical value to offer educators either at present or in the near future. Yet, it is our belief that work in artificial intelligence, both that which stresses the "artificial" (i.e., the development of machines which assist humans in the performance of a variety of problem-solving tasks) and that which focuses on "intelligence" (i.e., the exploration of human cognition, memory, and problem-solving behavior), has profound implications for the educational process. AI-based instructional programs, so-called intelligent tutors or coaches, have already been developed in some areas of the school curriculum, most notably in mathematics.

But it is the search for a deeper and more accurate understanding of how the mind works, the research on human intelligence, which promises the greatest dividends for education. Artificial intelligence offers an arena in which to test theories of mind. Simplistic models of how people store and

recall information or go about solving problems, for example, will no longer suffice; a researcher attempting to use an inadequate theory to write a program will run into problems which he or she will be unable to resolve without first revising that theory. As Schank and Hunter (1985) suggest, "that's why we need to write programs. Programming forces us to be explicit, and being explicit forces us to confront the problems with our theories." This sustained effort at building detailed process models of human cognition will perhaps prove to be AI's greatest contribution. In education, in particular, a better understanding of how people learn and solve problems, as well as a better grasp of what constitutes effective teaching, may turn out to be a far more significant outcome of AI research than any instructional program.

After more than a quarter of century of research and development, the field of artificial intelligence remains something of a mystery. Discussions about the educational uses of AI, in particular, have been afflicted with vague statements about the technology's potential, and with the kind of hype that had once accompanied the introduction of educational television and the first wave of computer-assisted instruction. One possible explanation for this state of affairs is that so few educational applications of AI have moved beyond the research laboratory.

Nonetheless, the moment seems ripe for an examination of AI and its educational applications. We already understand a great deal both about the design of expert systems--the representation of human expertise in a computer program--and about human cognition. In addition, a significant number of Intelligent Computer-Assisted Instructional (ICAI) systems have already been developed, which makes comparison and evaluation of these systems possible.

We define AI in the conventional manner, as "the branch of computer

science devoted to programming computers to carry out tasks that if carried out by human beings would require intelligence" (Graham, 1979). AI techniques are now being used for a broad range of tasks: problem-solving; natural language processing; perception and pattern recognition; information storage and retrieval; control of robots; game playing; automatic programming; and computational logic. ICAI systems make use of a number of these capabilities. Existing tutoring programs are able—in a limited way--to generate and solve problems, store and retrieve data, play games, diagnose student misconceptions, select appropriate teaching strategies, and carry out a natural-language dialogue with a student.

This paper has four major aims:

- to provide a description of ICAI systems and their components;
- to examine recent developments in ICAI research;
- to evaluate the potential impact of ICAI for school practice and for our understanding of learning and teaching; and
- to serve as a basis for discussions among educators, researchers, and policy makers regarding the uses of ICAI in schools.

Historical Context

If there is a major theme in the history of computer-based instruction, it is the trend from the rigid behaviorism of traditional CAI toward the more learner-oriented, cognitive approach of current ICAI (O'Shea & Self, 1983). Programs based on Skinnerian principles of operant conditioning have given way to tutoring systems that are sensitive to individual differences among students. In one important respect, however, both old and new CAI are driven

by the same vision: to provide effective instruction by utilizing the computer's capacity for immediate feedback and individualization.

The paper focuses on only one of three major educational applications of AI research. Indeed, ICAI is the least well known and the least widely used of these three approaches. One strategy, exemplified by LOGO, provides students with an environment in which they can explore a variety of programming strategies. The theory underlying this "learning by doing" approach is that students develop general problem-solving skills by being involved in programming; it is the activity of programming, rather than the specific subject of the program, that is the central focus of this approach. Drawing its inspiration from the ideas of Piaget, this strategy stands in direct opposition to classical CAI and its behaviorist assumptions.

The second approach uses games and simulations as instructional tools. As in LOGO, students are provided with a microworld in which they are expected to exercise their knowledge or skills, but here the instructional environment is more constrained: its limits are defined by the subject matter of the specific game or simulation. The approach is based on the theory that students gain an understanding of mathematical or scientific principles not by memorizing a body of knowledge, but by participating in a computer-based activity--say, a simulated prey-predator experiment. Participation typically involves manipulating a set of variables and observing their effect on a simulated real-world system (in this example, the relationship between prey and predator populations).

Unlike the LOGO and gaming approaches, CAI makes an explicit attempt to instigate and control the learning process. Although CAI programs may use simulations and games, their main focus is not on these activities, but on the instructional process itself. The goal of CAI research has been to build

carefully designed instructional programs that can be optimized to the needs of each student.

Traditional CAI: Linear and Branching Programs

The first generation of computer-based instructional programs were heavily influenced by Skinnerian ideas; they were based on the principle that the main task of teaching is to reinforce desired behavior (or successive approximations to desired behavior) (O'Shea & Self, 1983). These so-called "linear" programs had three main features. First, program output was organized into discrete frames each designed to take the student a small step toward the desired behavior. Second, students advanced through the program by responding to each frame of output, usually by filling in a blank. Third, the program reacted to the student's input by presenting the next frame, whose content was predetermined by the program author. Although the emphasis was on feedback and individualization, feedback was provided only after correct responses, and individualization meant only that a student could proceed at his or her own pace, not that the material was presented in a different way to different students.

Branching programs were an attempt to escape the rigidity of the linear approach and to use the student's response to control what the student saw next. The view of the student as a passive recipient of information remained unchanged, but the emphasis now was on the exposition being sensitive to the student's response. Branching programs differed from linear programs in a number of important ways. First, output frames tended to be larger, since there was less emphasis on the need for the student to always be correct. Second, branching programs were able to adapt, albeit in a limited way, to the needs of individual students: less able students were given more help than

abler students. An incorrect response resulted, for example, in an exposition of remedial material; the problematic frame was then repeated before moving on to the next frame in the curricular sequence.

Despite these differences, however,linear and branching programs have many features in common. Both emphasize the importance of systematic presentation. Both treat the student as "tabula rasa." Both are concerned with the issue of instructional efficiency and see learning as the acquisition of knowledge rather than experience. Both encourage students to do what is expected of them and provide no room for student initiative. Finally, both are instances of "programmed learning:" all branching decisions as well as the content of each frame are prespecified by the author. Although the branching strategies of some programs became quite involved, incorporating the best learning theory available, critics have argued that branching programs do not make effective use of the computer, and that many CAI systems are little more than programmed texts. Also, for complex subject matter, the difficulty of pre-specifying all possible instructional sequences becomes prohibitive.

Two Demonstration Projects in CAI: TICCIT and PLATO

In 1971, after more than a decade of research and development efforts in computer-assisted instruction, the National Science Foundation decided to explore the effectiveness of this instructional approach by investing ten million dollars over five years in two demonstration projects, known as TICCIT and PLATO. The aim of the TICCIT project (Time-shared Interactive Computer Controlled Information Television), which was carried out by MITRE Corporation, a major defense contractor, was to demonstrate that CAI could provide better instruction at less cost than traditional teaching methods in community colleges. The program was designed as a primary source of

instruction, and not merely as an adjunct to conventional teaching (O'Shea & Self, 1983).

Each student was provided with a terminal, which included a keyboard, a color television, and videotapes. A loudspeaker on the terminal provided prerecorded audio messages to the student. The student communicated with the system through the keyboard or a light pen. TICCIT was tested in pre-calculus and English composition courses at two community colleges.

Evaluation findings indicate that the system's emphasis on lowering per student costs made it unpopular with the faculty of the two community colleges. In addition, the development of course material, particularly in English composition, resulted in lengthy disputes over content and method. Furthermore, the effort required to develop software was grossly underestimated. Test results show that students who completed TICCIT courses attained higher scores than the control group. But the program suffered from very low completion rates. In math, the completion rate was 16 per cent compared to 50 per cent for non-TICCIT courses. In general, the system seems to have favored good students over poor students, and it was ineffective for those who could not manage their own instruction (O'Shea & Self, 1983).

The second of the NSF-funded projects was based on the University of Illinois' PLATO system (Programmed Logic for Automatic Teaching Operation). PLATO began with one terminal in 1960; by 1971, the NSF project (called PLATO IV) had about 950 terminals in 140 sites, and 8,000 hours of instructional materials contributed by more than 3,000 authors (O'Shea & Self, 1983). Its aims were to demonstrate the feasibility of a computer-based educational network; to demonstrate that such a system was economically viable and could serve any educational level; to develop curriculum materials; and to gain

acceptance by instructors and students. PLATO's implicit aim was for the system to grow into a national and even international network.

The project made use of very large networks of terminals and the latest in computer technology. Unlike TICCIT, it made little effort to control what the various program authors wrote either in terms of quality or content, and as a result the material produced was of variable quality. Teachers were permitted to use the system in whatever way and to whatever extent they wished. Students interacted with the program by means of a terminal and a plasma display panel (the panel had internal memory; it was also transparent, enabling slides to be projected and to be superimposed on computer-generated graphics). Students could also communicate with the system by using a touch screen.

The evaluation of PLATO IV showed mixed results, mainly because of the variation in quality among programs. As in TICCIT, the preparation of material turned out to be more difficult than anticipated. Unlike TICCIT, however, the system had drop-out rates no higher than for human instruction. Moreover, it was found to be generally popular with both student and teacher users. One reason why teachers liked it, despite the absence of clear-cut performance advantages, was because they retained control over how it was used (O'Shea & Self, 1983).

Early Uses of AI: Generative Programs

The requirement that the entire instructional process, including the content of each frame, be prespecified by the author makes the task of writing computer-based materials extremely time-consuming. "Generative CAI"--so called because of its ability to generate problems from a large database--came out of a desire to make the task of writing instructional programs

easier. It also emerged from a different educational philosophy, which was based on the belief that students learn better when they work on problems that are appropriate to their level of skill than when they follow a prespecified sequence of material.

Generative programs represent the earliest application of AI techniques in education. The approach has been used primarily in developing drill and practice programs in arithmetic and vocabulary recall. A generative system consists of three main components. The first is a set of parameters on a particular problem (for example, finding the equation of a line), which are then replaced by randomly generated integers in the course of an instructional session. The second is a link connecting problem types to appropriate solution processes (for example, the above problem may require finding the y-intercept). The third is a teaching strategy, which usually includes ordering problems according to their level of difficulty and knowing when to move from one problem to another.

The major advantage of generative CAI is that it can provide as many problems as the student needs to learn a skill, and can control the level of difficulty of the problems. Another advantage is that the courseware can solve the problems it generates and present step-by-step solutions to them. The program can ask relevant questions of the student (for example, in connection with a graphing problem it can ask, "What do you think the y-intercept is?"). It is also capable of answering problems generated by the student so long as the problems follow a format the system has been programmed to understand. Finally, generative programs are able to create student models based on stereotyped characterizations of student behavior. These are only rough estimates rather than precise indicators of students' knowledge; because of the simplicity of arithmetic and vocabulary drills, however, this

technique has proven sufficiently robust for instructional use.

Although generative CAI marks an advance over branching programs, it has two important limitations. First, this approach is limited to very simple knowledge domains. Its most effective application, for example, has been in arithmetic drill and practice. Second, few subjects are so well structured that they can be cast into a generative model, which requires a way of determining the relative difficulty of the material to be taught. Thus, the approach is not very useful for teaching subjects that admit to more than one correct answer, or where it is difficult to sort questions by order of difficulty.

Mathematical Models of Learning

The first efforts at building adaptive CAI, exemplified by branching and generative programs, were criticized for their reliance on informal theories of learning. In addition to being severely limited in their ability to adapt to the needs of different students, these programs had no way of knowing which teaching strategy would be most effective in any given situation. In an attempt to predict the effects of alternative teaching strategies, researchers built instructional programs that contained explicit representations of learning theories. These theories were expressed in mathematical notation.

To determine what teaching strategy is optimal, a program first needs to estimate the student's level of knowledge. One can then use this model of the student's knowledge to predict the effect of presenting a particular chunk of material. A teaching strategy can be individualized to a particular student by using the student's responses to learn about his or her prior level of knowledge and learning processes (style, speed, areas of difficulty).

The problem with mathematical models of learning is that, with the exception of a few special cases, we do not understand learning processes well enough to model them in mathematical form. In addition, mathematical or stochastic learning models tend to oversimplify the process of learning. Mastery of a subject usually involves gradually increasing comprehension of fuzzy concepts, whereas mathematical representations tend to view the acquisition of knowledge as a more black-and-white, linear process.

Recent Developments: Dialogue-based Tutorial Systems

Programmed learning and its implementation on computers in the form of branching programs led to regimented instructional approaches: the computer asked all the questions, and the student selected the answer from a small set of possible answers. Much of the subsequent development of CAI was a reaction to this regimentation. Generative CAI and mathematical models of learning were alternative strategies which provided more flexibility and adaptability, but their utility has proven to be limited because they presuppose well-structured problems and the existence of fully developed models of learning.

Although there is no sharp boundary between generative CAI and ICAI, the latter is characterized by a finer-grained capacity for student diagnosis, and a more sophisticated notion of tutoring and the varieties of teaching strategies that can be used. Current ICAI continues to use generative techniques to create new problems as they are needed in an instructional session.

Recent work in ICAI has attempted to increase student control over the machine. Researchers have developed learning environments which go beyond classical CAI to incorporate "learning by doing" approaches. These systems

combine games, simulations, or programming with machine-based tutors which are capable of engaging the student in a dialogue. Much of current ICAI research can be seen as an attempt to reconcile two apparently conflicting objectives. On the one hand, researchers have tried to develop good tutors whose ability depends, to a large extent, on limiting the student's instructional paths and exercises to those that can be completely specified ahead of time. On the other hand, they have tried to build programs based on "learning by doing" notions, which allow students freedom to explore, to make mistakes, to pursue dead ends--behaviors that cannot be completely specified in advance (Sleeman and Brown, 1982).

To be able to provide effective tutoring in a relatively open-ended environment, a system must have expertise in solving problems, the capability to diagnose or model student behavior, and the ability to intervene appropriately. In the following section we describe in some detail the components that make up a complete--that is, idealized—ICAI system. Even the best of the existing intelligent tutors, however, are only approximations of this ideal type; one must keep this fact in mind when reading descriptions of actual ICAI programs in Section III. Some have diagnostic capabilities, but lack a tutor; others are able to model student behavior and select appropriate teaching strategies, but have no natural language processing capabilities.

In abandoning behaviorism for a more cognitively-oriented approach, current research has also forsaken traditional CAI's objective of providing complete machine-based courses in the style of TICCIT or PLATO. Instead, developers have concentrated on programs of more limited curricular scope that provide support for, and serve as adjuncts to, conventional teaching methods.

Subject areas in which intelligent tutors have been developed include basic arithmetic (BUGGY, WEST); simple algebraic equations (LMS, O'Shea's

Quadratic Tutor); logic and problem-solving (WUMPUS); debugging of electronic circuits and computer programs (SOPHIE, LISP TUTOR); and medical diagnosis (GUIDON). Research efforts are currently under way in physics, organic chemistry, arithmetic, geometry, and probability theory.

SECTION II: DESCRIPTION OF ICAI SYSTEMS & THEIR COMPONENTS

In its ideal form, a complete ICAI system comprises four major components. The first component contains the substance of what is to be taught--the program's knowledge of a domain and its problem-solving expertise. This expertise module is capable of generating problems and comparing the student's solutions with optimal problem-solving strategies. The second component is responsible for diagnosing the student's knowledge of, and misconceptions about, the subject matter contained in the system's knowledge base. This student module indicates what the student knows and does not know and constructs a model of the student which is used to guide the instructional dialogue. The third component specifies how the system presents material to the student, to eliminate misconceptions and to communicate new knowledge. This teaching module synthesizes knowledge about natural-language dialogues, teaching methods, and subject matter; selects problems for the student to solve; monitors his or her performance; provides assistance upon request; and selects remedial material. The fourth module handles communication between student and system, typically by using some method of natural language processing.

Expertise Module

The type of knowledge representation scheme used in a system's database will depend on the domain in question as well as on the purposes for which the program has been designed. An expertise module about geography, for example, will typically consist of a large set of facts about objects in the domain (so called "declarative" knowledge): information about the region and countries in question, capitals and other major cities, languages, populations,

agricultural and industrial products. In addition, actions and events with their time sequence and causal relationships can be modeled in a declarative format. Such a representation scheme, for example, has been used to teach about the causes of rainfall (see WHY in section three).

A second type of subject-related knowledge, however, may also be required. Expertise requires manipulating facts in the database to draw inferences, solve problems, and perform a variety of other tasks. This so called "procedural knowledge" is what humans use to ride a bicycle, or compose sentences, or do arithmetic operations. In highly structured content areas such as arithmetic or electronics, procedural knowledge is contained in a set of definitive rules about what to do and when to do it. In other, less well-specified domains, however, procedural knowledge consists largely of "rules of thumb" about how to proceed in an ambiguous or uncertain environment. Such heuristic rules are the kind of knowledge which is usually absent from textbooks, but which enable a domain expert to make a decision even when he or she is uncertain of the facts or when the evidence points to more than one answer.

Some computer programs also make use of knowledge about knowledge ("meta-knowledge"), which has to do with the system's ability to assess what it knows, the extent and origin of its knowledge about a particular subject, and the reliability of a given piece of information. SCHOLAR, for example, can reason plausibly about some aspect of South American geography without having any explicit knowledge of the subject in question (see section three).

Knowledge representation is the most active area of AI research today. Modeling the knowledge of human experts in a computer program lies at the heart of the research on expert systems. The construction of these knowledge-based systems is typically done by sitting down with human experts, watching

their work, and asking questions about how they make decisions. Because these experts are not always able to provide answers that fit easily into a computer program, some way has to be found of representing their knowledge: the rules and procedures they use in their work. Researchers look not only at declarative knowledge but also for procedural knowledge: how a particular scientist makes a particular kind of decision under conditions of uncertainty.

The knowledge inside an expert's head is largely heuristic knowledge, based on experience and couched in uncertainty; expertise is often more a matter of good guesses than of facts and rigor. Moreover, much of the expert's knowledge is private because he or she is unable to share it with others; frequently, an expert is not even aware of all he or she knows. But if someone undertakes patient observation of the expert in the act of doing what he or she does best, the knowledge can be teased out and made explicit. This painstaking method, called knowledge engineering, has been applied in a variety of domains, including organic chemistry, the diagnosis of infectious diseases (see the discussion on GUIDON in section three), geological exploration, and elementary and advanced mathematics.

Another part of research on knowledge representation concerns the modeling of this human expertise in a program. For a time, AI had been split between those who preferred declarative representation schemes and those who thought procedural representations were more effective. "Declarativists" have stressed the static aspects of knowledge--facts about objects and events and the relationships among them. Proponents of procedural schemes have argued that AI systems have to know how to use their knowledge: how to find relevant facts and make inferences. Declarativists have pointed to the flexibility and economy of their representation schemes and to the modularity and modifiability of their systems. Proceduralists have talked about the

directness and transparency of reasoning that their programs facilitate. The earliest AI programs tended to use one representation approach at the exclusion of the other. More recently, researchers have attempted to combine declarative and procedural strategies in an effort to exploit the strengths of each approach.

Types of Representation Schemes

The major task confronting AI researchers has been to create programs that exhibit intelligent behavior. Toward that goal, researchers have developed schemes for incorporating knowledge about the world into their programs; these involve routines for manipulating specialized data structures to make intelligent inferences. Each scheme for representing knowledge touches on issues central to the study of cognition and intelligence. Although there is a great deal of debate about the strengths and weaknesses of various representation schemes, very little is known with any certainty about why certain schemes are better than others for particular knowledge domains. In the following paragraphs, we describe some of the principal methods used to represent knowledge in ICAI systems.

One of the first declarative methods for representing knowledge, "state-space search," was developed to allow programs to play games such as chess. The search space is not so much a representation of knowledge as it is a representation of the structure and rules governing the game. This includes knowledge of the available alternatives at each stage of the problem--for example, the possible legal moves at each turn of a chess game. The scheme is based on the idea that, from a given state in a problem, all possible next states can be determined with a small set of rules (in chess, these correspond to the rules for moving each piece).

The problem is that with the most interesting problems (like chess), far too many possible combinations of moves are possible; state-space search leads to "combinatorial explosion." A major goal in AI research has been to limit the number of alternatives examined at each stage of a problem-solving sequence to the best available choices. In order to determine what alternatives are best, the program must make use of large amounts of knowledge, encoded within the program in some knowledge representation. In general, whatever the domain, the goal of research in knowledge representation has been to allow AI programs to behave as if they knew something about the problems they solve.

A second widely-used declarative representation scheme is based on formal logic. The advantage of a formal approach is that it makes use of a set of rules, called rules of inference, which allow the derivation of new information from facts that are already known to be true. Logic-based schemes also allow the truth of any new statement to be checked against what is already known to be true. The most important feature of these systems is that deductions are guaranteed to be correct (given accurate assumptions), a level of certainty that other representation schemes cannot match. One weakness of such schemes, however, is that when they are used with large databases, a combinatorial explosion can sometimes result. A logic system may have trouble selecting which rules to apply to which facts at each step of a complex proof.

A third declarative scheme, semantic nets, was developed as a psychological model of human associative memory. A semantic net consists of nodes (which represent objects, concepts, and events) and links between the nodes (representing their interrelations). One key feature of semantic net representations is that important associations can be made with a minimum of search through the database: relevant facts about an object or concept can be

traced directly from the nodes to which they are directly linked. It is both a weakness and strength of semantic nets that, in contrast to logic-based representation systems, they use a wholly content-dependent reasoning strategy. While there is no inherent logic to reasoning based on nets, the reasoning strategy that is used in a net representation has the benefit of being directly familiar to humans.

The second major category of knowledge representation schemes is characterized by data structures in which knowledge about the world is contained in procedures--small programs that know how to do specific things. Procedural representation techniques, for example, have been used to create parsers for natural language communications systems; parsers use grammar rules and other information to determine the functions of words in a sentence. Procedural techniques have also been used to represent a wide variety of skills; SOPHIE, for example, uses a procedural approach to represent the expertise of an electronic troubleshooter (see section three). Despite their powerful problem-solving capabilities, however, procedural representation schemes have a major flow: the underlying domain knowledge is organized in a manner that is not readily accessible to humans. As a result, it is difficult to verify and change programs that use procedural representation.

The production system approach is an attempt at addressing the limitations of a procedural representation and at linking declarative and procedural knowledge. First developed by Newell and Simon (1972) as a model of their theory of human cognition, it allows for a modular representation of knowledge. This approach is finding increasing popularity among developers of large AI programs, in which the database consists of rules, called

productions, which specify a particular action given certain preconditions. These rules take the general form: "If this event occurs, then do this action."

The usefulness of the production system approach derives from the fact that each rule and the conditions in which it applies are stated in terms that are understandable to humans. In addition, interactions between rules are minimized (that is, one rule does not call upon another). Production systems have been useful in controlling the interaction between statements of declarative and procedural knowledge. Moreover, because they facilitate human understanding and the modification of programs with large amounts of knowledge, production systems have been central to building systems that contain large databases such as MYCIN (see section three).

"Frames and Scripts," another knowledge representation scheme, is still in its formative stages of development. A frame is a data structure that includes both declarative and procedural information in predefined internal relations. An interesting feature of this scheme is the ability of a frame to determine whether it is applicable in a given situation. The system may select a frame to aid in a given situation; this frame tries to match itself to the data it discovers, but, if it finds it is not applicable, control is transferred to a more appropriate frame. Scripts are frame-like structures specifically designed for representing sequences of events (see WHY in section three). Both scripts and frames refer to methods of organizing the knowledge representation that facilitate recall and inference.

Ultimately, all knowledge representation schemes are interchangeable; if we know one scheme in sufficient detail, we can construct an alternative. It is the intended use of the knowledge by the computer program that recommends one representation scheme over another. Indeed, some AI researchers recommend

the use of multiple representations of the same information as a strategy for creating truly effective and versatile tutorial systems (Stevens, Collins, and Goldin, 1982).

Articulate Expertise

Domain expertise can be either "glass-box"--that is, a program can explain each problem-solving decision it makes in terms that a human can understand--or "black box"--a program can use data structures and algorithms that do not mimic those used by humans. Historically, most expert systems have used the latter approach, largely because a black-box expert is not constrained by human-like algorithms and hence tends to be more efficient in performing the kinds of large-scale, complex computations for which knowledge-based systems are typically used.

But a tutoring system requires glass-box capabilities in order to be effective. The capacity for human-like reasoning can be used both for the evaluation of the student's moves and for determining the reasoning underlying these choices. Skills can be determined by looking at the expert's problem-solving behavior and noting the processes involved. A glass-box expert is also useful for evaluation because it can generate all alternative "better" moves and hence determine the rank ordering of a given move. However, the ability to evaluate the student's moves, which involves determining the complete range of alternate moves, requires much more computation than simply assessing the skills underlying a particular move. A black-box expert tends to be more efficient for such computations, but, since the skills it employs are not analogous to those of a human, they cannot be used for categorizing the student's skills.

One solution to this dilemma is to combine an efficient and robust

black-box expert for evaluation with a less efficient glass-box expert for skill determination. This is the strategy used by the developers of WEST (see section three); they constructed a black-box expert for evaluating the student and augmented it with small pieces of an articulate expert which were used to identify those places in the instructional process where tutoring was appropriate. These pieces of articulate expertise enable WEST to work backward from a solution to determine what parts of the problem-solving procedure require tutorial intervention.

The Student Module

The student-model module is a recent focus of ICAI research; one measure of its importance is that it serves as the major theme of Sleeman and Brown's Intelligent Tutoring Systems (1982). The module is used to hypothesize about the student's misconceptions and less than optimal problem-solving behavior. The goals of the module are to predict student behavior in learning situations and to indicate the causes of student errors. The information contained in the student model is used by the tutoring module to point out misunderstandings and faulty strategies, indicate why they are wrong, and suggest corrections.

The student model is constructed by posing questions to the student and analysing his or her responses. In addition, ICAI systems use "flags" in the system's knowledge base to indicate the areas mastered by the student. Information used to maintain the student model includes pupil problem-solving behavior, direct questions asked of the student, assumptions based on the user's experience, and assumptions about the difficulty of the material.

Researchers have used a variety of techniques to construct student

models. In this section we describe three major constructs for representing student knowledge: the "overlay" model, which views the student's knowledge as a subset of an expert's knowledge (Goldstein, 1982); the "differential" model, which focuses on the differences between the behavior of a student and that of an expert (Burton & Brown, 1982); and the "pertubation" model, in which the student's misconceptions are viewed as variants of correct problem-solving procedures (Burton, 1982).

The overlay model is perhaps the most commonly used of the student modeling schemes; it has been used, for example, in SCHOLAR and SOPHIE (see section three), as well as in a number of math tutors. This model views expertise as a set of facts or rules, and the student's knowledge is represented as a subset of this knowledge. Tutoring consists of encouraging the growth of this subset by intervening where a missing fact or rule is needed to arrive at a correct solution.

The overlay model is constructed from hypotheses about which skills of the machine-based expert the student possesses. The system hypothesizes that the student does not possess a skill if the student's answer to a given problem is worse than the expert could have provided using that skill. This hypothesizing is based on a two-part calculation: first, for each problem-solving situation, the system records how many times a particular skill should have been used (based on its model of expert behavior); then, the system records how many times the student used the rule under the appropriate circumstances. The ratio of these two numbers defines the frequency with which the skill was used; when the ratio exceeds a certain cut-off point, the program assumes that the student knows the rule.

The principal shortcoming of the overlay model is that it fails to account for the way in which new knowledge and problem-solving skills usually

evolve--major means of new knowledge acquisition include analogy, generalization, debugging, and refinement. The model also fails to consider that a novice may not be using a subset of the expert's skills, but rather his or her own rule constructs, which may bear little resemblance to those of the expert.

The "differential" model was first designed as the diagnostic component of a computer-based game (Burton and Brown, 1982) and was intended to address the special problems that a gaming environment poses for effective instruction. Perhaps the main reason that a game is problematic as an instructional environment is that little explicit diagnostic activity can occur: the coach or tutor cannot use tests or pose multiple questions without interferring with the flow of the game. Instead, the system must remain unobtrusive, restricting itself to making inferrences about the student's weaknesses from his or her behavior while playing the game.

This diagnostic strategy poses difficulties for an ICAI system. The absence of a demonstrable skill may not always signify ignorance of that skill; it may simply indicate that the student has had no occasion to use it. The absence of a skill in a sequence of problem-solving activites has diagnostic importance only if an expert in the same situation would have used it. To cope with the restrictions imposed by a gaming environment, Burton and Brown have devised a "differential" modeling technique that compares what the student is doing with what an expert would have done under similar circumstances. The "difference" in their behaviors provides the basis for making hypotheses about the extent of the student's knowledge and the nature of his or her misconceptions.

The construction of a differential model requires two steps: evaluating the student's behavior in relation to the moves an expert might have made

under similar conditions, and determining the underlying skills that went into the selection of the student's action as well as the better moves of the expert. For the evaluation of the student's behavior, the system needs only the results of its reasoning strategy--that is, a black-box expert can be used. For the second task, however, the system needs access to the knowledge that went into the selection of the better moves--glass-box expertise is required.

A third approach to modeling student behavior comes out of the research on BUGGY and DEBUGGY (see section three). Called a "pertubation" or "buggy" model, this strategy views student errors as systematic "bugs" in the repertoire of the student's problem-solving behaviors. The pertubation model is an attempt to improve upon the usual kind of diagnosis, which consists of determining whether or not a student has mastered a particular skill and perhaps also the degree of mastery of the skill. This typical diagnostic approach offers little help in the instructional process, because it is not fine-grained enough to specify which parts of the skill need improvement. Without a model of the skill being tested, it may even be difficult to determine if a student possesses the skill (Burton, 1982).

The more precise kind of diagnosis that is made possible by the pertubation approach involves, first, specifying the set of subskills in a given skill and, second, determining which of those subskills are missing from the student's repertoire of skills. The model is able to reproduce the student's behavior on problems that the student has already worked and to predict his or her behavior on future problems. Indeed, its developers expect the model "to be able to predict, not only whether the answer is incorrect, but the exact digits of the incorrect answer on a novel problem" (Burton, 1982).

The pertubation model has to deal with two potential sources of uncertainty. On the one hand, there may be more than one bug that accounts for the student's answer. On the other, there may be no systematic bug at all that has caused the student's error; mistakes may be due, for example, to "performance lapses"--fatigue, laziness, boredom, or inattention. The system must be able to find the student's systematic errors even when non-systematic errors are present.

The perturbution model's current solution to the presence of noise is to include only those bugs that account for at least forty per cent of the student's errors. Bugs that do not meet this condition are rejected; those that remain become part of the diagnosis. Each bug is then classified on the basis of how well it explains the student's answers.

The Teaching Module (will be added later)

The Communication Module

A sentence can be analyzed either from a syntactic or a semantic point of view. Syntax, or grammar, specifies rules for combining words in a certain order, and allows us to analyze these word patterns without paying attention to their meaning. The sentence, "I been to the supermarket," for example, is syntactically incorrect but meaningful; we know that the word "been" should be accompanied by an auxiliary verb such as "have" or "had," but we have no problem understanding what the statement is trying to say. Sometimes, however, syntactical mistakes can interfere with the meaning of a sentence. The sentence, "The been supermarket to I," not only looks grammatically incorrect but also sounds like nonsense.

By contrast, semantics is concerned with the meaning of words or sentences as they are used in a particular context. A sentence may remain syntactialy unchanged, but alter its meaning in different circumstances. For example, "I see the picture" can mean one thing in the context of a visit to an art gallery and quite another as a response to an explanation. Although syntactic concerns are important in language understanding, it is this ability to handle the semantic complexity of a language such as English or French that ultimately distinguishes the expert user of language from the novice, the native from the beginning foreign student.

Similarly, the major obstacles to improved natural language processing by computers have more to do with the semantic complexity of language than with syntax. At present, the limited ability of knowledge-based systems to understand natural language--which is based on the program's use of grammar rules as well as domain-specific knowledge stored in its database--is more

akin to that of a first-year foreign student than to that of an expert native speaker.

In natural language processing, grammatical rules are used to parse input sentences, that is, break them apart to get at their meaning. Several different kinds of grammars have been used in natural language programs, including the semantic grammar scheme used in SOPHIE, which substitutes the usual categories of noun, verb, and adjective for more domain-specific characterizations of the words used (see section three). Parsing consists of using grammar rules to determine how words are used in an input sentence in order to build a data structure (sometimes this structure is thought of as a diagram) representing the relations among words in a sentence. This structure can then be used to get at the meaning of a sentence. All natural language processing programs contain a parsing component. The design of a parser is a complex problem. In addition to having to specify the grammar to be used, a decision must also be made about the method of use of the grammar--that is, the manner in which word sequences are matched against patterns of the grammar.

Knowledge-based Natural Language Systmes

By the early 1970s, researchers had developed systems that attempted to deal with both syntactic and semantic aspects of language processing. These programs integrated syntactic and semantic analysis with knowledge about a limited domain, allowing them to deal with more sophisticated aspects of language than previously possible. The increased sophistication of these programs had a great deal to do with advances in knowledge representation. Indeed, research in natural language understanding has been closely connected with development efforts in knowledge representation (Barr & Feigenbaum, 1981).

Both procedural and declarative knowledge representation schemes have

been used to create natural language processing systems. Perhaps the most widely used declarative approach has been semantic networks, in which words and their meaning are represented as a set of linked nodes, a data structure that facilitates the drawing of inferences (see SCHOLAR in section three).

Case representations extend the notion of semantic nets with the idea of a "case frame"--the clustering of properties of an object or event into a single concept. One variation of this theme is Roger Schank's idea of "conceptual dependency" (see below). The advantage of case representations is that they group relevant sets of relationships into single data structures. The idea of clustering structures has been used in representation schemes based on the notion of frames. In analyzing an input phrase or sentence, a frame-based language understanding system tries to match the input to the prototypes of objects and events in its domain that are stored in its database.

Researchers have used frame-like data structures called "scripts," which represent sterotyped sequences of events, to understand simple stories. This approach assumes the events being described will fit within one of the program's scripts, which the program will use to fill in missing pieces in the story. The common element in all frame-based systems is that frames make possible the use of expectations about the properties of objects or events--about what typically happens in a variety of familiar situations. Frame-based programs compare an ambiguous sentence with what would be expected based on a prototype. If there is a plausible match, assumptions can be made about what is meant (Barr & Feigenbaum, 1981).

Schank and Hunter illustrate their notion of "expectation packages" with a story about a man who goes to a restaurant, orders a sandwich, and leaves a large tip because the waiter brought the sandwich quickly. Although the story

does not state explicitly that John ate the sandwich and payed for it, these actions are implied by the context. "When we hear about a restaurant, we expect to hear about a variety of objects, events, and people. There should be a menu, the patron should look at the menu, pick something, tell the order to a waiter, or waitress, wait for a while, be served, eat the food, have the table cleared, get a check, pay the check, leave a tip, and depart" (Schank & Hunter, 1985).

The investigation of script- and frame-based systems is the most active area of AI research in natural language understanding. A major focus of current research is the construction of a communication module that can duplicate the human ability to understand a partial utterance or sentence that relies for its meaning on implicit references to previous statements. These are common features of everyday discourse, but they present enormous problems for any machine-based language processor. For example, "Where is the Empire State building?" "How tall is it?" both refer to the Empire State building. To interpret such anaphoric references, the system must use information gathered from earlier statements; in this case, the program must link the pronoun "it" in the second sentence to the subject in the first sentence.

Similarly, elliptic references require filling in incomplete phrases using terms already mentioned. For example, "How tall is the Empire State building?" "... the Sears building?" are both questions concerned with the height of the buildings. When we hear questions such as these, we implicitly fill in the missing parts; it is an activity that comes so easily to us that we take it for granted. But it is precisely this kind of common-sense skill in language understanding that a computer program finds extremely difficult.

Conclusion

No ICAI system has completely satisfactory models for each of the four components discussed in this section. For one thing, understanding natural language is an extremely complex task. Natural language systems are not as flexible as human speech is, requiring users to restrict themselves to a subset of the language--a limited number of vocabulary words and syntax rules. Also, we do not know very much about differences in how people reason or learn; for this reason, the problem-solving strategies in the expertise model may not be appropriate for all users, particularly nonadults. In addition, expert tutoring systems are not really skilled at pedagogy compared to human instructors. Unlike a good human teacher, the intelligent tutor simply follows a small set of instructional rules and is unable to take into account a student's physical and verbal cues, affective style, or level of interest.

SECTION III: EXAMPLES OF ICAI SYSTEMS

The ICAI tutors described below illustrate a variety of knowledge representation schemes, diagnostic models, and teaching strategies. These programs exemplify some of the major directions taken by early ICAI research as well as some of the strengths and weaknesses of existing intelligent tutoring systems. Their approximate dates of completion are given in parentheses; multiple dates indicate different versions of the same program.

SCHOLAR (1970, 1973, 1975)

SCHOLAR, a pioneering effort in the development of computer tutors, is capable of handling unanticipated student questions and of generating instructional material in varying levels of detail. The program was created by Jaime Carbonell, Allan Collins, and their colleagues at Bolt Beranek and Newman, Inc., to instruct students in South American geography. An example of a "mixed-initiative" tutor, the system allows both student and machine tutor to control the dialogue. Like any competent human tutor, the program can take the initiative during an instructional dialogue to question the student about the extent of his or her knowledge or to determine the nature of the student's misconceptions.

SCHOLAR's mixed-initiative capabilities are based on its ability, within limits, to process a restricted form of natural language: both the student's input and the program's output take the form of English sentences. The system can handle unanticipated student questions as long as they correspond to a list of expected question types contained within the program. Also, SCHOLAR can produce short, simple English sentences that contain no clauses and that make use of only a few verbs.

The program's knowledge base is represented in a semantic net. As noted in Section II, this representation scheme is an attempt at modeling the human capacity to form associations among objects or concepts. SCHOLAR organizes information so that relevant facts about a topic can be inferred from the nodes to which that topic is directly linked. Knowledge can thus be stored effectively for fast, easy retrieval. Each node or unit in SCHOLAR's knowledge base corresponds to some geographical object or concept (composed of a name and a set of properties associated with that name).

SCHOLAR was one of the first attempts to explore the characteristics of Socratic dialogue and to model this teaching strategy in a program. The tutor elicits responses from the student, uses these responses to diagnose misconceptions, and then presents material to encourage the student to revise his or her faulty knowledge. The program is based on an analysis of human tutorial dialogues and attempts to mirror the behavior of expert human tutors.

SCHOLAR employs a variety of inference strategies--production rules--for responding to student questions and for evaluating student answers. These strategies are designed to cope with the incompleteness of the information stored in SCHOLAR's knowledge base and allow the program to reason about the extent of its knowledge—to engage in "plausible reasoning" (Collins, 1978). For example, one such strategy is called "intersection search" and enables the program to answer questions such as "Is Buenos Aires in Argentina?" The system undertakes a search attempting to find an intersection linking Buenos Aires with Argentina. If no intersection were to exist, it would answer, "No." In this instance, however, it finds an intersection linking the node for Buenos Aires with the node for Argentina. Indeed, the semantic net is organized in such a way that the program can distinguish between the question above and a similar question in which subject and object are reversed: "Is

Argentina in Buenos Aires?" To the latter question SCHOLAR is able to answer, "No, Buenos Aires is in Argentina" (Barr and Feigenbaum, 1982).

A second inference strategy makes use of the notion of open and closed sets. Sets that contain all relevant objects are defined as "closed," while sets that contain some but not all the objects in a given category are defined as "open." This labelling enables the system to determine when its knowledge is incomplete and to reason about the extent of its knowledge. In SCHOLAR's semantic net, sets of objects which satisfy some condition (e.g. countries on the Pacific) are labelled as either open or closed.

The program can arrive at a plausible answer to a student's question about rubber production in Guyana, for example, without having any explicit knowledge of the subject (Barr and Feigenbaum, 1982). SCHOLAR will look into its knowledge base, compare Guyana's agricultural products with the agricultural products of countries it knows produce rubber, infer that it knows as much about Guyana as it knows about the other countries, and infer further that it would know about rubber production in Guyana if it were important. The program will then conclude, though somewhat uncertainly, that rubber is not produced in Guyana.

SCHOLAR will provide the answer about Guyana, explain its reasoning, and indicate a lack of certainty about its conclusion. The program will also provide the student with whatever information it has about rubber production in other countries in South America. SCHOLAR's ability to reason about the extent of its knowledge represents an application of meta-level knowledge (knowledge about knowledge).

WHY (1976, 1978)

A tutor on the causes of rainfall, WHY is an extension of the SCHOLAR research. Like SCHOLAR, WHY is based on an analysis of tutorial dialogues, but here the content deals with something more complex than simple declarative facts about geography. Rainfall is a complex geophysical process involving the interaction of numerous factors, including temperature, wind, land and water masses.

In developing WHY, researchers focused on three principal issues: the characteristics of a good tutor, the nature of student misconceptions, and effective strategies for explaining complex processes (Stevens, Collins, and Goldin, 1982). By analyzing tutorial dialogues between human experts and students, the program's developers were able to identify some of the elements that make for good pedagogy. They found, for example, that students learn to reason most effectively about complex processes by first working on specific problems and then trying to generalize from them. They also discovered that Socratic dialogue was especially effective for instructing students in domains that require complex reasoning (Collins, 1976).

A major feature of WHY's Socratic method is its ability to evaluate a student's response (say, a mistaken hypothesis about the causes of rainfall in the Amazon region of South America) and to select a counterexample that challenges the student to revise his or her theory. The goal of WHY is to develop in the student a "causal model" of rainfall: to enable the student to answer questions, give explanations, and make predictions about the causal relations involved in producing rain.

In WHY, the causal model of rainfall is represented as a script-like sequence of events. The script for heavy rainfall, for example, consists of four steps--evaporation, movement of air mass, cooling, and precipitation.

These steps are connected sequentially in the following manner: warm air absorbs moisture from a body of water, then winds carry the moist air over land, which causes the moist air mass to cool, which in turn causes precipitation.

In addition, WHY utilizes subscripts which provide more detailed explanation for individual steps in the main script. The process of evaporation, for example, is described in a subscript that contains a five-step sequence: 1) a body of water is warm; 2) this enables moisture to evaporate rapidly into the air; 3) in addition, the air mass over the water is warm; 4) this, in turn, enables the air mass to hold a lot of moisture; 5) these conditions--warm air and warm water--enable the air to absorb a large amount of moisture from the water.

The program prompts the student to suggest causes of rainfall, to look for prior or intermediate causes and, finally, to suggest a general rule. When such a rule is proposed, the system finds a counterexample (if the proposed rule is, in fact, faulty) and challenges the student to revise the rule to account for it.

A major finding of the WHY research is that the type of representational formalism used in a program determines the kind of tutoring that is possible, as well as the program's ability to handle questions and diagnose student misconceptions. For example, although WHY is adequate for representing misconceptions that result from missing or extra steps in its script-like knowledge structure, many other kinds of misconceptions cannot be diagnosed by the program. The WHY representation scheme emphasizes the sequential, temporally-oriented aspects of the rainfall process, but is inadequate for handling other, equally valid perspectives on rainfall. Questions such as, "Does the temperature of water affect evaporation?" or "What happens to the

temperature of air as it rises?" do not fit into a temporal, sequential format, requiring instead a functional representation.

A functional view emphasizes the roles that different objects play in the operation of a complex system. As Stevens, Collins and Goldin point out,

> a functional perspective differs from the scriptal view in several ways. First of all, it is non-linear and interactive rather than ordered and sequential. Positive or inverse functional relations "work" in either direction, whereas "increase" and "decrease" would be encoded as different events in a script representation. Secondly, causal relations are implicit and indefinite in the causal view, rather than explicitly stated as in the scriptal structure. Saying that air humidity is positively related to air temperature suggests some causal relation, but does not spell out which factor is primary. (1982: 16).

A functional representation includes a set of actors (each with a role to play in the overall process); a set of attributes (which describe the actors and affect the process); the result of the process, which is always a change in the value of some attribute (e.g., the result of evaporation is to increase the humidity of the air); and the functional relationship (e.g., the positive relation between the temperature of the moisture source and the humidity of the air mass).

Aware of the limitations of WHY's script structure, Stevens and Collins explored the possibilities of adding a functional perspective. They devised a questionnaire containing such items as, "How is the moisture content of the air related to heavy rainfall?" and "What causes evaporation?" to test students' understanding of rainfall. They were able to identify sixteen common user "bugs," none of which had been diagnosed earlier under the

original script-like representation. Many of the bugs turned out to be specific to the domain of rainfall. For example, the most common misconceptions had to do with the cooling and heating of air masses. Thus, WHY is limited in effectiveness by its lack of a functional representation.

Analysis of human tutorial dialogues shows that tutors spend a lot of time diagnosing conceptual bugs that students make; have domain-specific knowledge of the types of errors students are likely to make; and are able to organize their knowledge in different ways to meet the demands of particular instructional settings. Perhaps the most significant findings of the WHY research is that an effective computer-based tutor, able to catch most student misconceptions and provide appropriate instruction, requires more than one representation of knowledge. In Section IV, applications of the knowledge gained from SCHOLAR and WHY to a more complex program (STEAMER) are discussed.

SOPHIE (1974, 1975, 1977)

SOPHIE (SOPHisticated Instructional Environment) was created by John Seeley Brown, Richard Burton, and their colleagues at BBN to provide a learning environment in which students acquire problem-solving skills by trying out their ideas rather than by direct instruction. These problem-solving skills are developed in the context of a simulated electronics lab. The students' task is to find faults in a malfunctioning piece of equipment whose characteristics they obtain by taking measurements--voltages, currents, resistances--to determine what is wrong.

The instructional system has a model of problem-solving in its knowledge base and includes heuristic strategies for answering student questions, criticizing hypotheses, and suggesting alternatives for current theories. SOPHIE evaluates hypotheses by considering all the information that a student

should have been able to derive from his or her current set of measurements and by constructing reasonable hypotheses from this derived knowledge base. The program also judges the merits of a student's suggestion for a new measurement in light of the prior sequence of measurements; SOPHIE can decide if the measurement is valuable (does it eliminate or isolate a fault?) and will inform the student when the proposed measurement provides no new information.

A major issue in ICAI-related research has been the development of a natural-language processing capability that allows the student to communicate easily with the system. Experts agree that students will quickly become frustrated if they must try several ways of expressing an idea before the system can understand. SOPHIE attempts to cope with linguistic problems such as anaphoric references, context-dependent deletions, and ellipses (see Section II), which occur frequently in natural dialogues.

SOPHIE's natural-language capabilities are based on the concept of "performance" or "semantic" grammar, in which conventional categories such as noun, verb, and adjective are replaced by semantically meaningful categories. These categories represent concepts known to the system--measurements, circuit elements, transistors, hypotheses. The grammar, built on semantic categories, allows the system's parser to deal with "fuzziness" or uncertainty.

If a student uses certain words or concepts that the system does not know, the parser can ignore these words or concepts and try to make sense of what remains. As a safeguard against any misunderstanding, the program responds to a question with a full sentence, thus indicating what inferred question is being answered.

SOPHIE performs several different logical and tutorial tasks. It answers hypothetical questions of the "what...if" variety and evaluates student

hypotheses (e.g. if a student's assertion is logically consistent with the data already collected by the student). The instructional program can generate its own hypotheses based on known information. Finally, the system can determine if a given measurement is redundant (could results of this test have been predicted given what was already known?).

Extentions of SOPHIE include a troubleshooting game for two teams of students and the development of an articulate expert debugger/explainer. The expert can not only locate student-inserted faults, but can also explain its strategy in diagnosing the problem.

One of the system's weaknesses is its inability to take an active role in correcting student errors. Since the program is designed to be reactive to the student, it cannot take the initiative to explore students' misunderstandings or suggest new approaches. Section IV discusses more recent research directed toward correcting this weakness.

BUGGY (1978) and DEBUGGY (1981)

More a diagnostic tool than a tutor, BUGGY was developed by John Seely Brown, Richard Burton, and Kathy M. Larkin at BBN to diagnose a child's misconceptions about simple arithmetic operations. Its diagnostic capabilities go beyond simply determining whether or not a student has mastered a skill; BUGGY is able to construct a detailed model of the student's knowledge of basic arithmetic.

The system is based on a theory of student errors that departs radically from the usual view of mistakes, namely, that these are caused primarily by careless or erratic behavior. Brown and Burton's (1978) alternative view can be called a "programming" model of student problem-solving behavior; they argue that students, like computers, are extremely good at following

procedures, but that sometimes there are faults in these procedures, just as there are bugs in computer programs. BUGGY enables teachers to improve their skills in diagnosing systematic errors in their students' work.

The program represents the skills of subtraction and addition as a collection of subskills (for example, knowing how to subtract a larger digit from a smaller digit). These subskills are written into the system as subprocedures and are linked together in a procedural net (discussed in Section II). If all the subprocedures in the procedural network are correct, then BUGGY will do subtraction problems correctly. If, on the other hand, the procedural network contains one or more faulty subprocedures, then the program will produce systematic errors.

A student model containing faulty subprocedures can be used as a diagnostic tool for discovering the bugs that cause a student's errors. The model can also be used to help train teachers in diagnosing student bugs by playing the part of a student with faulty subskills. When BUGGY is used to diagnose a student's errors, the program searches its knowledge base for every combination of correct and incorrect subprocedures which will account for all of the student's answers, both right and wrong. Modification of the original, correct procedural net is accomplished by systematically replacing correct subprocedures with incorrect variations until a consistent student model has been constructed.

The program currently contains 110 primitive faulty subprocedures for subtraction. However, some errors are the result of more than one simple bug. BUGGY research has identified 20 common "combination" bugs, consisting of two faulty operations; combinations of three or more bugs were found to contribute to "combinatorial explosion" (see Part II), were thought to be relatively rare in actual student performance, and were therefore ignored for purposes of the

program.

BUGGY can also be used to train teachers in diagnosing student errors. The program presents a teacher or a team of teachers with a series of mathematical errors and asks them to construct a theory about an underlying bug. Teachers are then given a number of problems to see whether they can predict the faulty answers a student with this bug would give. If the teachers' theory is inadequate to explain the series of errors, they can try again, constructing new problems for the program to solve and using these buggy "solutions" to reformulate their theory. When the teachers indicate they are ready to try out their revised theory, BUGGY provides them with a new set of problems on which to predict student errors. The program concludes that the correct diagnosis has been made when teachers provide it with at least five correct predictions.

The use of procedural networks to represent well-defined subject matter such as arithmetic facilitates the decomposition of skills into subskills and the construction of student models. In contrast to overlay modeling, which represents student skills as a subset of expert problem solving skills (see Section II), the procedural network approach views student behavior as a deviation from correct procedures. Such an approach seems justified by the types of mistakes students learning arithmetic operations typically make.

DEBUGGY, developed by Richard Burton at Xerox Palo Alto Research Center, is an extension of the BUGGY work. Based on their experience with BUGGY, Burton and his colleagues decided to investigate one procedural skill in depth rather than attempt to examine two or three subject areas simultaneously. The procedural skill they chose was multi-digit subtraction. The DEBUGGY diagnostic system is able to construct much more sophisticated models of student behavior than BUGGY (Burton, 1982).

DEBUGGY analyzes a student's solutions to a set of problems and places each student in one of four diagnostic categories. First, it grades the student's problem set. If the student makes no mistakes, he or she is placed in the "Correct" category. Then, the program searches for the set of bugs that best fits the student's errors. If the fit is good, the student is placed in the "Buggy" category; if not, the system must decide whether the student's errors are few enough to be diagnosed as unsystematic "performance slips." Alternatively, if there are too many errors, the system labels the student's behavior as "Undiagnosed," which indicates that a human teacher should intervene as diagnostician (VanLehn, 1981).

DEBUGGY has performed impressively as a diagnostic and research tool. Human experts sometimes disagree with its opinions, but no more than they disagree among themselves. However, it has proven to be less effective as anf educational instrument. For one thing, teachers are not prepared to use its diagnostic capabilities, in part because the concept of bugs is foreign to them. For another, building a database on subtraction bugs has been time consuming and difficult. A similar effort would be required for each new procedural skill--a daunting prospect even if one were to consider only the domain of basic arithmetic. (In Section IV, more recent research building on BUGGY and DEBUGGY is discussed.)

GUIDON (1979)

Designed to teach diagnostic skills to medical students, GUIDON is a tutorial version of MYCIN, an expert system that provides consultations on infectious disease diagnosis and therapy (Shortliffe, 1974). Like SCHOLAR, GUIDON is a mixed-initiative tutor; it plays an active role in choosing knowledge to present to a student, based on his or her competence and interests. The goal of its developers has been to study the problem of

transferring machine-based expert knowledge to learners(Clancey, 1982).

MYCIN interacts with physicians in much the same way that medical specialists do: it asks questions about the patient and provides advice about therapy. Its knowledge is represented in the form of conditional sentences, called "production rules," that provide information about what to do in a given situation. A principal feature of this type of representation scheme is the separation of the knowledge base from the interpreter that guides its use. This separation allows the system's knowledge to be used as both expert consultant and tutor.

In addition, the production rule formalism used in MYCIN can generalize to domains other than medicine. Because the knowledge base is separate from the interpreter, detailed rules about a new domain can be substituted for medical knowledge to create a new expert system with the same meta-knowledge. (The domain-independent package, consisting of rule interpreter and explanation module, is called EMYCIN, for "essential MYCIN." Its production rule formalism has been shown to work successfully in other domains, such as providing advice on structural analysis problems in chemistry, interpreting pulmonary function tests, and recommending drug therapy for psychiatric patients.)

Production rules provide a flexible and easily understood representation of facts in the domain and the relations among these facts. The following is typical of the approximately 450 rules contained in GUIDON's knowledge base: "IF (1) the gram stain of the organism is gram negative, and (2) the morphology of the organism is rod, and (3) the aerobicity of the organism is anaerobic, THEN there is suggestive evidence (0.6) that the genus of the organism is Bacteroides" (Clancey, 1982). Evaluations have shown that MYCIN's ability to treat meningitis and bacteremia is equal to that of the

infectious disease faculty at the Stanford University Medical School (although the program's ability to diagnose degrades rapidly outside of its narrow specialties).

In addition to MYCIN's expertise, GUIDON contains three other major components. First, the program has knowledge about dialogue patterns that enable it to understand the student and to generate utterances. Its dialogue capabilities are based on studies of discourse in AI (e.g. Winograd, 1977), which suggest that "there are places in a discourse where questions make sense, others where explanations are expected " (Bruce, 1975; quoted in Clancey, 1982).

Second, GUIDON has an augmented knowledge base consisting of frames (see Section II), annotations to the rules, and the factors used by rules. For example, the program contains canned-text descriptions of every lab test in MYCIN, including descriptions of how each should be performed as well as explanations of how a given factor leads to a particular infection. For example, the frame associated with the factor "a seriously burned patient," explains "that the organisms originate in the air and grow in the exposed tissue of a burn, resulting in a frequently fatal infection" (Clancey, 1982).

Third, the program has knowledge of the context in which communication takes place. One major component of this context has to do with the student's intentions and level of knowledge. GUIDON uses an "overlay model" to represent the student's knowledge, which it views as a subset of the expert system's knowledge.

A second major component of the communications situation consists of two interconnected modules: a "case syllabus" (a lesson plan of the topics to be discussed in each case), and a "focus record" (to keep track of the factors in which the student has shown interest). Such knowledge of the communications

situation is used to control the use of dialogue patterns by the program.

A student's input can take the form either of menu options or of simple English phrases that are parsed using keyword analysis and pattern matching. A tutorial session begins with a student describing the kind of case he or she wants to learn about, say, a "burned meningitis patient." GUIDON selects a case that meets this description and provides initial information. The student is then expected to ask questions in order to elicit further information.

If the student does not know how to proceed, he or she can ask for help. The program may respond, "Try to determine type of infection." Here the tutor has set a goal—"determine type of infection"—and will use it to evaluate the student's questions in this portion of the dialogue. The question, "What's the patient's white blood count?" will be judged to be relevant to the above goal, and the program will provide the requested data. If, on the other hand, the student's question is judged to be irrelevant, GUIDON will indicate this and suggest another line of questioning. The program's use of a "goal-directed dialogue" enables it to keep track of the student's performance as he or she solves the problem and to provide appropriate assistance.

The program is able to update the overlay model of the student. When the student proposes a hypothesis, GUIDON will ask a number of true-false, fill-in, and multiple-choice questions to evaluate the extent of the student's knowledge. Information gathered in this way is then used to revise the system's model of the student.

In contrast to some other intelligent tutors, GUIDON does not contain multiple forms of knowledge representation. Its developers concluded that, given the nature of the reasoning that is required in diagnosing and treating infectious diseases—students (and physicians) need to evaluate empirical

information, rather than construct arguments about causal processes--a rules-based representation scheme is sufficient for instructional purposes.

As a result, MYCIN provides "cookbook" responses to data and makes no attempt to explain data in terms of physiological mechanisms. Moreover, MYCIN's expertise operates in a "closed" world. Unlike WHY, for example, which can handle questions about things it does not know explicitly by reasoning about the extent of its knowledge, GUIDON tutorials are limited to one hundred cases; and all the data that are relevant to their solution are already contained in MYCIN's knowledge base.

Furthermore, in contrast to WEST (see below), which is able to rank each student response (or move) according to whether it is superior or inferior to other alternatives available to the student, GUIDON does not contain a model for ordering the collection of data. The reason for this omission has to do with the current state of medical problem solving. While medicine has conventions about the kind of routine data to collect, there is no consensus about how to order the search for information; medical diagnosis remains something of an art and, therefore, so does medical instruction.

The "intelligence" of GUIDON's tutoring component resides in its ability to select topics and to focus the instructional dialogue, both important characteristics of good teaching. How domain-independent its tutorial strategies are remains to be tested. (More recent work by Clancey on NEOMYCIN is discussed in Section IV.)

WEST (1978)

WEST is a coach for the children's game, "How the West Was Won," which was originally designed at Project PLATO (see Section I) to give students drill and practice in arithmetic. The program attempts to integrate an

exploratory learning environment--a game--with a coach, in order to explore the uses of coaching as an instructional strategy, particularly in situations where most of the control resides with the student rather than the teacher.

In contrast to the more structured setting of tutorials, games provide an informal, flexible, and open-ended environment for children to exercise their problem-solving skills; whatever learning takes place comes as an indirect consequence of the child's play. One major benefit of learning within the context of a game is motivation; a child may then carry over his or her newly-acquired problem-solving skills to other domains.

The open-endedness of an instructional game, however, does present problems for the teacher. For example, the instructor must pay close attention to student choices in order to steer players away from gross errors, to help them perceive their misconceptions and sub-optimal tactics, and to suggest alternative strategies. Also, the game must be designed in such a way as to allow students to exercise their initiative and control, to make demonstrable mistakes, and to learn from their mistakes.

For a game to be an effective learning activity, an expert advisor is needed who acts as guide and observer, who can explain student errors and suggest more effective strategies, and who knows when to intervene and what to say at the appropriate occasion. This is the type of role usually assigned to a coach. WEST is a program designed to play such a coaching role. Central to its pedagogical approach, which can be described as "guided discovery learning," is the notion that one of the most important aspects of a learning environment is the degree to which the student is allowed to learn from his or her mistakes.

The creation of WEST required solutions to two major problems: When doesa coach interrupt the student? What should a coach teach when intervening?

WEST contains both the capability to construct a student model and a set of tutoring principles that provide guidance for interrupting and advising a student.

The program's ability to construct a diagnostic model of the student is constrained by the gaming environment: WEST cannot make use of prestored tests or pose diagnostic questions (since these would interfere with the flow of the game). Instead, the program must infer the student's misconceptions and weaknesses from whatever he or she does while playing. This diagnostic strategy, however, results in some ambiguity. When a student fails to employ a particular skill, whether this is due to the student's ignorance or to some other reason is not always clear. Thus, the absence of a skill has diagnostic value only if an expert would always have used it in a similar situation.

WHY performs its diagnosis by comparing the student's behavior with what its computer-based expert would have done in the student's place. In order to carry out this strategy of "differential modeling," the program must evaluate the student's move in relation to a set of possible alternative moves the expert might have made. WEST must also determine the underlying skills that went into the selection of the student's move as well as each of the better moves of the expert so that advice can be given which will improve performance.

WEST employs a black-box representation of its knowledge domain in conjunction with a set of "local" glass-box experts (see Section II for a discussion of glass-box and black-box experts). The black-box expert is used to determine the set of possible moves that the student could have made, and the glass-box mini-experts help to diagnose the causes of the student's less than optimal behavior. As a rule, an articulate expert, which can explain each problem-solving decision it makes in terms that a student can understand,

makes for a better coach than an inarticulate expert. However, a glass-box expert is often less efficient than a black-box expert in evaluating student moves, an activity which requires a great deal of computation. WEST augments the more efficient and robust black-box expert with small pieces of an articulate expert which are used to indicate when some intervention is justified.

The skills and concepts a student is expected to master determine what parts of the student's behavior are monitored by the coach. Each skill represents a portion of the articulate expert and has an associated recognition and evaluation component. The recognizer is used to watch the student and create a model of his or her behavior; the evaluator uses a comparison between expert and student models to decide if the student is weak in that skill. If it is determined that a particular skill is lacking, the coach presents an explanation together with a better move that illustrates that skill's use.

The program uses some fifteen pedagogical principles to guide its coaching behavior. Perhaps the most important of these principles is that which specifies four distinct levels of hints when a student requests aid. In response to the first request for help, the coach locates a missing skill that is required at this particular point in the game and suggests its use. The second request for help causes the program to provide the student with a complete set of possible moves. The third request for help results in the coach selecting the optimal move and explaining why it is optimal, while the fourth request causes the coach to describe how to make the optimal move.

The sophistication of WEST is limited by the simplicity of the game within which the coach is embedded. Further work by Burton and Brown has centered on more complex learning situations (see part IV).

GEOMETRY TUTOR (1985)

The Geometry Tutor is one of two ICAI systems currently under development by John R. Anderson and his colleagues at Carnegie-Mellon University (the other is the LISP Tutor). The intent of the Geometry and LISP Tutor research is to understand how students arrive at solutions to problems, and what the major difficulties are in developing problem-solving skills, in order to develop a tutoring system that helps students acquire these skills. The tutors are being used to test a general theory of skill acquisition and to establish a set of guidelines for designing tutoring systems in other domains that also require problem solving (Boyle and Anderson, 1984).

Anderson and his colleagues base their work on the observation that private human tutors are much more effective than either group classroom instruction or standard CAI; indeed, their findings indicate that tutoring is at least twice as effective as the other two approaches (Anderson, Boyle, Farrell, and Reiser, 1984). Where the Carnegie-Mellon group differs from some other ICAI researchers is in its approach to the modeling of tutoring expertise. Stevens, Collins, and Goldin (1982), for example, apply expert systems techniques to the building of tutoring programs. They view the human tutor as an expert whose knowledge can be extracted and represented in an instructional system. This is the technique used in both SCHOLAR and WHY.

Anderson and his colleagues argue that human tutoring is too diffuse a skill to be subjected to knowledge engineering techniques. Unlike other better specified domains, there is too much variation among tutors--and no crystallized expertise--to be modeled. Much of ICAI research shares Anderson's view: instead of trying to mimic an actual human tutor, researchers have tried to take principles of effective tutoring from instructional theory and to embody these principles in machine-based tutors.

The GEOMETRY TUTOR is based on a detailed cognitive model of how students solve problems and learn. The system attempts not only to solve geometry proofs, but tries to do so the way a successful student would. This "ideal" student model consists of a set of production rules which recommend particular rules of inference when specific conditions prevail.

The tutor uses a differential student modeling approach: expert and non-expert behaviors are represented as two distinct sets of production rules. When a bug occurs, the system measures the difference between the rules used by the ideal model and those used by the student to determine the student's weaknesses. The system then presents remedial material intended to overcome these deficiencies. The goal of the instruction is to reduce the differences between the expert's set of rules and those used by the student.

The system is guided in its tutorial strategy by the student model and a set of eight instructional principles. Among the most important of these principles is that effective problem-solving behavior, including the construction of proofs, involves the decomposition of major goals into subgoals and an iterative two-directional search for solutions--forward from a set of initial conditions to the final goal, as well as backward from the proposition to the givens. Other principles guiding the tutor's behavior are that students learn better if instruction takes place in a problem-solving context; that immediate feedback on errors keeps the student from wasting a lot of time pursuing wrong strategies; and that problem solving is facilitated if the student is not required to keep in mind every step of the solution.

The system has been shown to be effective in preliminary tests with a few students of varying abilities. For the good student, the program provides a way of keeping track, on the screen, of all inference steps taken as well as the capability for catching performance slips. More importantly, the tutor

can provide the poor student with help by suggesting inference strategies when he or she does not know how to proceed. The system can respond either to a request for help or to an inappropriate use of a rule by means of pop-up windows. These windows display applicable rules and their definitions. In addition, when the student makes a mistake, windows are used to explain why it was a mistake. The system can interrupt the student while he or she is making an error, suggest more appropriate inference steps, and question the student to make certain the step is understood.

Many of the design issues surrounding the construction of the Geometry Tutor have had to do with human factors: where to position material on the screen; what abbreviations to use; when to correct misspellings; how to let the student point; how to design the student-system interface; how to relate proof structures to the diagram that accompanies the proof. Solutions to these problems have had very little to do with theory and a great deal to do with trial and error. It was found, for example, that the tutor is more effective if the student first spends an hour or so practicing on the system a more familiar area such as arithmetic. The student can thus learn how the system works without also having to worry about new subject matter.

The major limitation of the approach used by the Geometry Tutor (acknowledged by its developers) is that it requires a domain in which an ideal student model can be completely specified. That is, only in such areas as high school and early college math, and introductory programming, can we determine ahead of time exactly what the student is supposed to do.

Conclusion

The main focus of early ICAI research was on developing the basic techniques for building intelligent tutors: finding effective strategies for

representing expert knowledge, for understanding the student's strengths and weaknesses, for selecting appropriate teaching strategies, and for communicating between user and machine. When this research effort was coupled to a serious instructional application the program was typically limited to a domain outside the K-12 curriculum -- for example, electronic engineering (SOPHIE) or medicine (GUIDON). The programs that dealt most closely with the conventional school curriculum had to do almost exclusively with basic arithmetic operations, as in WEST and BUGGY. Other programs, such as SCHOLAR and WHY, which covered non-arithmetic topics, were intended more as explorations of what constitutes good tutoring than as direct contributions to the elementary or secondary classroom.

Current ICAI research is paying much closer attention to the K-12 curriculum. In addition to Anderson's work in geometry, programs now under development include tutors in fractions (Lauren Resnick, Learning Research and Development Center-University of Pittsburgh), physics (Joan Heller and Frederick Reif, University of California, Berkeley), and organic chemistry (Jill Larkin, Carnegie-Mellon University). (Much of this work is too preliminary to be reviewed here.) At the same time, researchers are continuing to focus on a wide range of ICAI issues, and are using prototype programs as laboratories for basic research on knowledge representation, the nature of meaning and cognition, and the characteristics of good tutoring.

SECTION IV: ASSESSING INTELLIGENT COMPUTER-ASSISTED INSTRUCTION

So far, this study has presented an overview of ICAI to date. The historical evolution of the field has been delineated, the nature of ICAI discussed, and examples of typical systems described. The next two sections offer a forward-looking perspective: present directions of research, capabilities and limitations at the current state of the art, and long range implications for schools. The intent is not to present an exhaustive review and synthesis (which would be beyond the scope of this paper), but instead to indicate illustrative emerging issues in AI-based educational tools.

MAJOR THEMES IN CURRENT RESEARCH

In section two of the study, four major components of ICAI systems were described: modules incorporating the knowledge base, student model, pedagogy, and communications. Separation of functions is one of the major design principles in AI, since this facilitates individualization, modification, and generalization. Of course, the boundaries created are to some extent arbitrary; in ICAI, all four components blur together in the actual process of teaching. Within each of these overlapping areas, what types of research are now in progress?

Expertise Module

How can one enable a device to understand what it is teaching? The problem is more complex than giving the computer a set of interconnected concepts and skills—though that is difficult enough. Because AI is the study of "dynamic" intelligence (the process of intellect), the ways in which the

machine structures its conceptualization (metacognition) are as important as the content of its thinking.

In ICAI, this issue poses significant challenges, because the computer must reflect human-like thought patterns if students are to understand the cognitive strategies it models. The expertise in an instructional technology must be "articulate." Ideally, even if the tutor or coach actually solves problems using "black box" (cognitively opaque) methods for reasons of effciency and robustness, the device also would have a "glass box" (transparent to people) reasoning mechanism in its knowledge representation. Thus, what a computer knows, how it thinks about its knowledge, and the extent to which its cognitive processes are comparable to those of humans are all important.

Another major aspect of knowledge-based system design is that "in the knowledge lies the power" (domain-specific information and skills are more crucial to expert problem solving than general concepts and reasoning strategies). This principle implies that the knowledge base for each specialized subject must be tailored, incorporating detailed ideas and processes rather than simply a broad theoretical orientation and generic inference tools. This has made the ICAI development process time-consuming, as--for each narrow domain--explicit representations of expert cognition must be conceptualized and programmed into specialized coaches and tutors.

An ICAI expertise module, while necessarily narrow in its content domain, serves a variety of functions. Educational applications demand a knowledge representation which facilitates access, reasoning, planning, problem solving, pattern recognition, communication, acquisition/expansion, hypothesis generation/evaluation, and question answering. No single method of encoding content can maximize all these capabilities--each purpose might have a different

optimal representation--so multiple strategies for incorporating information are needed in a single program.

Unlike an expert system, educational applications also require a mix of declarative (what), procedural (how), and metacognitive (thinking about what and how) knowledge. Historically, semantic nets and schema-based coding strategies have been used for descriptive representation, while production systems, rule-based methods, and simulation approaches have modeled process information. Finding ways to combine the strengths of both these representation types in an overall metacognitive structure is a major field of research within knowledge based systems in general. Tools such as object oriented languages (e.g., SMALLTALK) and truth maintenance (constraint-based) systems are facilitating work toward this goal (Borning, 1981).

All these themes have implications beyond ICAI and are currently being explored from a variety of AI perspectives. Three illustrative areas of research focus on the nature of expertise, the transfer of meaning, and the sequencing of knowledge.

Characteristics of Expert Cognition

Being expert in a limited domain involves the following characteristics: doing things most practitioners cannot, being smooth and efficient, using specialized knowledge and methods, and having rapid information-sorting skills. Researchers studying expert performance have noted that the cognitive processes involved seem to be as much recognition as reasoning (Clancey, 1984). Based on years of practical experience, the expert rapidly relates the current situation to a pattern previously encountered. Only when the immediate problem seems novel does he or she fall back on theoretical concepts and inductive strategies (Larkin, 1980).

In the early stages of learning a field, many experiences are new and must be thought through. With time, patterns of association begin to develop and, in the expert, eventually become "compiled" (indexed into condition/action pairs via chunking of perception as an aid to short term memory) (Shiffrin and Dumais, 1981). These contextually stimulated pattern responses become linked to behavioral scripts that the expert follows.

Expertise becomes an increasingly automatic, intuitive process of rapid access to specialized, interconnected information. Skills in situation recognition, standard problem solving, and dealing with novel developments are necessary. These require a mixture of deductive, schematic, and heuristic knowledge (Johnson, 1982).

In knowledge engineering (transferring skilled problem solving from person to machine), this automaticity of expertise is a major challenge. The "authentic" knowledge of experts is "tacit," since compiled, intuitive cognition is difficult to communicate to others. Often, what the expert can verbalize is "reconstructed" knowledge (similar to the expositions given in textbooks). These general rules of thumb, while useful, do not convey the ability to make rapid decisions in a particular situation.

Authentic knowledge can be studied by monitoring how an expert approaches a "veridical" task: a problem representative of situations actually encountered in the field. (This would contrast with a "nonveridical" task, such as the structured problems at the end of a textbook chapter.) The goal of the knowledge engineer becomes to synthesize reconstructed knowledge (which an expert reports) and authentic knowledge (from observational tracing of the expert's thought processes) into "mechanically optimized" representations for computer problem solving (Johnson, 1984).

Some researchers have studied the incorporation of domain-dependent

expertise into knowledge representations. EXCHECK uses inference procedures characteristic of professional mathematicians to understand proofs (Blaine and Smith, 1977). Standard problem solving approaches can be made more comprehensible to students by incorporating in the expertise module a formal language unique to that subject material which describes the main steps in the algorithm being used (Sleeman, 1977). Studies of how skilled readers form a complex inferential model to understand textual material can facilitate the development of sophisticated comprehension programs (Collins, Brown, and Larkin; 1979).

The inclusion of metacognitive skills in the expertise module has also been researched. Systems such as SCHOLAR (discussed earlier in part three) include imprecise, incomplete, and uncertain knowledge. This enables the use of sophisticated inferring and problem-solving approaches (Collins et al., 1975).

Context-free problem solving grammars, based on taxonomies of planning and error correction, provide a language for communicating useful strategies (Miller, 1982). Kimball (1982) and O'Shea (1982) describe tutors capable of self-improvement in subject-domain understanding through interaction with students. Control knowledge of goals and tasks is important in implementing any metacognitive strategy (Anderson, 1983).

Research on the nature of expertise has generated a number of questions important in developing ICAI systems. While "black box" portions of the knowledge representation can utilize mechanically optimized cognitive strategies, what mixture of reconstructed and authentic reasoning should be in the "glass box" model? How can the tacit knowledge--both domain dependent and metacognitive—of experts most efficiently be incorporated into an instructional knowledge base? What level of procedural justification must be

incorporated in an expert problemsolver to produce explanatory capabilities? The XPLAIN (Swartout, 1983) and NEOMYCIN (Clancey, 1983) systems are two interesting approaches to resolving these issues.

To what extent should students be trained to be experts (as opposed to practitioners), and what does this say about the mixture of deductive, schematic, and heuristic knowledge they need? If expertise is ultimately based on years of practical experience, how much can a "learning by doing" AI-based tool foster such skilled performance? What mixture of veridical and nonveridical tasks should be used in transferring expertise from tutor or coach to student? These illustrative questions indicate the range of ICAI issues raised by the study of expert cognition.

The Nature of Meaning

Researchers believe that the learning, recall, execution, and adaptation of complex procedures are enhanced by conveying multiple levels of meaning. Four illustrative dimensions on which a process can be understood are 1) mental models of how it works, 2) the tasks and goals it accomplishes, 3) the strengths and limits of its user, and 4) its larger context of related procedures. Together, these types of meaning form a web of semantic rationalizations that is the foundation of understanding (Brown, 1982).

Mental models of how a system works represent its parts, functions, and interactions. Comprehending common sense physical reasoning requires qualitative thinking about processes, their effects, and their limits (Forbus, 1983). "Envisioning" a situation in this way aids decisionmaking by forecasting future events through an underlying conception of how the system functions (de Kleer and Brown, 1983).

People seem to use qualitative causal reasoning to envision systems; this

integrates structural, functional, and constraint-based types of models. Such an approach combines the benefits of detailed mechanistic analysis, teleological (designed purpose) clues (Van Lehn and Brown, 1980), and inference from global behavioral limits (Kuipers, 1982). The SOPHIE system's knowledge representation utilizes this type of sophisticated reasoning model (Brown, Burton, de Kleer; 1982), as does the STEAMER simulation (Stevens, et al., 1981).

Understanding the causal order of events allows comprehension first of the basic behavior of subsystems, then of the system as a whole. When the situation being studied is complex, people tend to adopt simplified versions of qualitative causal reasoning. Using "rationalized hypothetical reconstruction," a "kernel model" that reflects the basic behavior of a system gradually is modified to predict more complex behaviors as sophisticated features are added.

Mental models promote an understanding of the connection between structure and function in a system, as well as making assumptions explicit and building up a knowledge of how the system behaves. Procedural skills related to the system then can be applied with full awareness of their causal meaning and a sense of their underlying theoretical justification. This is an important type of understanding for a knowledge base to contain.

The tasks and goals accomplished in a system by a complex procedure supply a second dimension of meaning (Genesereth, 1982). Understanding the goal structure of a task helps users comprehend how the subprocedures within a process are accomplishing portions of the task and why the overall procedure follows a specific temporal pattern. Also, the systemic context imposes constraints on how tasks can be completed, giving a rationale for why the procedure has a certain structure. In addition, "boundary tasks" (barely

achievable in the system) indicate the limits within which a procedure may be used. (The metacognition references discussed in the "nature of expertise" section give more detail on artificial intelligence approaches to issues of goals, planning, and control.)

Human strengths and limits provide a third dimension of meaning for complex procedures. Size constraints on short term memory impose a cognitive load that can slow down execution of a procedure, so processes are often designed to minimize the data that the user must track internally. Mnemonic (memorable) patterns of organizing procedures provide a means for ensuring that all parts of the process are remembered. Wickelgren (1976) and Tulving (1983) provide overviews of cognitive retention issues.

With practice in using a procedure, the user will remember previous errors, developing an "event-based semantics" (expertise as recognition) to reduce mistakes. Physical contraints on human performance also shape procedural design. These cognitive factors provide an important context for understanding why a process has a certain structure.

Finally, the procedures to which a given process is related supply an overall framework of meaning. Learning and usage are enhanced if a procedure is linked to others already understood. Error checking, prevention, and recovery portions of a process connect its purpose to the larger operational and social context (Brown, 1982).

The web of semantic rationalizations created by these four illustrative dimensions of meaning has numerous implications for ICAI. What mix of types of meaning is optimal in transferring understanding of a complex procedure from tutor or coach to student? How can "learning by doing" be designed to

build envisioning of a system and qualitative causal reasoning about its dynamics? What types of "kernel models" would be most useful in aiding instruction?

How can metacognitive expertise on tasks and goals for a complex procedure be accumulated and incorporated into knowledge representations? What types of cognitive research on human strengths and limits are needed? How should the learning of processes be sequenced to maximize links to previous procedures and interconnection into an overall framework of semantic rationalization?

Sequencing of Knowledge

The design of an ICAI system's expertise module is influential in determining the nature of its student and pedagogical models. The user of a coach or tutor often is simulated via subsets, simplifications, or deviations of the expert's knowledge. Similarly, the teaching module's choice of information to convey is limited to items from the knowledge representation. How these pieces of knowledge are sequenced in instruction is determined by the developmental level and current comprehension of the student, by the type of teaching method being used, and by the evolutionary structure of information within the expertise module.

This evolutionary structure forms a "syllabus" of knowledge from which the tutor or coach can select (Goldstein, 1976). Subsets of the expert's knowledge are sequenced to reflect progressive difficulty and prerequisite constructs. For example, "scripts" which link knowledge into functional explanations are the approach used in WHY to creating syllabi (Stevens, Collins, Goldin: 1982). Metacognitive rules for simplification within the knowledge representation can aid in creating intermediate learning

environments (similar to the kernel model discussed earlier under meaning) and in summarizing complex explanations. The goal is to provide the learner with "frontier knowledge" which builds on the edges of what is currently understood.

"Genetic graphs" offer one research approach to incorporating syllabi in the expertise module. The term comes from Piaget and uses "genetic" in the sense of "source and growth"; he thought of himself as a genetic epistemologist (the origin and development of knowledge). Expert behaviors can be analyzed in terms of tasks, which may be conceptualized as a curriculum-like series of topics. As novices accumulate experience and wisdom, their knowledge moves along this network of increasingly sophisticated strategies for problem-solving. A "model" path representing the natural progression of an average learner can be postulated (diSessa, 1982).

The links which connect these clusters of problem-solving approaches may be thought of as evolutionary relationships (Goldstein, 1982). One set of strategies may suggest another via generalization/specialization, analogy (Douglas and Moran, 1983), deviation/correction, or simplification/refinement. Such a theory builds a bridge between expert-based and learner-based paradigms for ICAI, combining the developmental and cognitive attributes of the student with the intrinsic characteristics of the knowledge base.

The incorporation of syllabi into ICAI expertise modules raises a number of questions. How can "relative difficulty" and "prerequisite" be determined for different populations of learners? To what extent are simplification rules domain independent? What diagnostic information can be gained from individual deviations off the model path? How can the metacognitive skills represented by evolutionary links in the graph best be developed? Much research in cognitive science is needed to resolve these issues.

Conclusions

Regardless of interest in ICAI, studies on ways to represent knowledge in computers will continue; this area is linked to the commercial possibilities of expert systems as well as to many fundamental theoretical issues in artificial intelligence. Research in knowledge-based systems goes far beyond the range of illustrative topics covered above.

However, the development of expertise modules for computer coaches and tutors depends on more than general advances in knowledge-based systems. The questions delineated earlier are specific to educational applications of artificial intelligence and will require targeted research and funding for their resolution.

Student Model

Early ICAI programs adapted to individual learning situations by using stereotypic models of the pupil. As the power of computers and of artificial intelligence techniques has increased, the goals of student modeling have become more ambitious: prediction of the learning behavior of individual users and diagnosis of the causes of errors. These goals require an internal model of the learner, which must represent cognitive processes (such as information retrieval, calculation, and problem solving), metacognitive strategies (e.g., learning from errors), and psychological attributes (developmental level, learning style, interests). In building this learner representation, the intelligent coach or tutor uses four types of evidence: implicit (from student behavior in problemsolving situations), explicit (based on dialogue between ICAI devices and pupil), structural (from intrinsic complexity relations among knowledge representation skills), and background (based on estimates of average learner proficiency) (Carr and Goldstein, 1977).

A student model of this sophistication necessitates an even more complex representation than the simulation of expertise discussed earlier. Psychological capabilities and style must be added to cognitive and metacognitive skills, and the full spectrum of mental configurations on these three dimensions over the user population must be incorporated. While some of the challenges this poses can be resolved by modeling the student as a subset, simplification, or deviation of the expert's knowledge, other aspects require extensive study of learner characteristics and new representational approaches.

Rich (1983) gives three dimensions on which user models can be classified:

- --a model of a single stereotype-user versus a collection of models of individual users
- --models specified by the user or systems designer versus models inferred by the system based on the user's behavior
- --models of long-term user characteristics versus models of the current task

In addition, the nature and form of information contained in the user model and the inference engine needed to interpret that information constitute a fourth important dimension (Sleeman, 1984a).

All ICAI systems incorporate a collection of user models based on inferred behaviors, so fall into the second category of the first two dimensions. The other two dimensions are useful in differentiating types of intelligent tutoring and coaching programs. In this paper, however, the focus will be on illustrative areas of research in the nature of student knowledge, errors, and learning.

Student Knowledge

If a coach or tutor can compare the pupil's understanding of the subject

to its own expert representation, then instruction can center on transferring the missing or distorted portions of the knowledge base to the learner. One approach to student modeling is to visualize the user as an "overlay" (subset) of the articulate expertise in the ICAI program. The internal representation of the learner becomes a set of hypotheses regarding the student's mastery of the knowledge base.

As the user responds to instructional questions and situations, conclusions about the pupil's current understanding can be drawn. The accuracy of these hypotheses about comprehension can be tested by diagnostic initiatives which compare the student's responses to those which the expert module would generate given only the knowledge in the pupil subset. The genetic graphs discussed earlier under knowledge sequencing are a type of overlay approach (Goldstein, 1982). In the UMFE system, implementor-defined inference rules are used to determine, with minimum intrusion, which concepts the user knows (Sleeman, 1984a).

"Differential" models are an alternative theoretical construct. Here, "recognizers" within the ICAI program abstract and summarize the student's behavior in instructional learning situations, comparing the skills demonstrated to the expertise module's responses under identical circumstances. In this way, the issues in the subject domain the student does not understand can be determined through inference from differential weaknesses between pupil and expert. Sleeman and Hendley (1982) use a comparison algorithm based on formal language statements (such as those described earlier under domain-dependent expert representations) to create this differential model.

An important theoretical issue here is the "apportionment of blame/credit" (Brown and Burton, 1982). If a student does not make the best

response, which missing issue was the cause of this weakness? (The discussion of WEST in part three of this paper illustrates this point.) Because the coach does not intervene via diagnostic situations as in the overlay model, "noise" from student learning, inconsistency, or ambiguity makes construction of a differential model more difficult.

As with all constructs which model the user in terms of the expert, if the learner is employing a different problem solving strategy altogether, the student representation must incorporate a way of responding to this alternative approach. "Perturbation" models (which address this issue by representing the student's misconceptions as deviations from the correct skill) will be discussed below in the section on student errors. One problem for all these types of models is "combinatorial explosion": many alternative explanations for a given sequence of student responses.

Numerous questions for ICAI arise from research on student knowledge. What characteristics of the subject domain, learner population, and teaching strategy are influential in determining which type of student knowledge model to use? Without producing an overwhelming number of possible explanations, how can these different approaches be mixed to combine their strengths? Can all alternative student strategies for problem solving be captured by combinations of these types of representations? How can "noise" be reduced in the diagnostic process while keeping the measures used unobtrusive?

Student Errors

Beyond creating a "perturbation" model for user knowledge, research on the cause of pupil mistakes is vital in determining the best instructional strategy to correct misconceptions. Studies of student errors using protocol analysis are one means of better understanding this area (Putnam et al.,

1984). For each domain, a taxonomy of mistakes can be defined; a student's error pattern may then be used to diagnose missing concepts and processes in his or her knowledge base (Steven, Collins, Goldin; 1982). Interaction between errors complicates this type of analysis (parallel to the "apportionment of credit/blame" problem discussed above).

Students extrapolate "base rules" (derived from instruction and from previous problem solving experience) to new situations. This adaptation may occur through reconceptualizing unfamiliar problems into standard formats or by revising rules to be applicable in new cases. Systematic types of errors stemming from faulty revision and reconceptualization can be identified by protocol analysis. The instructional response can then be directed both toward correcting the error and toward examining the extrapolation process responsible for the faulty adaptation (Matz, 1982).

A "perturbation" construct of "bugs" (systematic modifications in correct skills) is one theoretical approach to modeling the results of such protocol analysis. A procedural network to mimic learner behavior is build into the student module. By creating a taxonomy of erroneous procedures (reflecting the smallest possible deviations in accurate strategies from the expertise module), a variety of "buggy" skills can be defined.

Faulty skills are then substituted for correct procedures until the learner model predicts student mistakes in detail. This diagnosis of learner bugs (and interacting combinations of bugs) indicates where remediation is needed to substitute accurate problem solving strategies (Burton, 1982). (The discussion of BUGGY in part three is an example of this modeling approach.

Young and O'Shea (1981) use an alternative, production system method for predicting student errors. They believe that pupil mistakes can be modeled via a combination of correct problem solving being omitted and rules from

unrelated domains being included. This viewpoint is closer to an overlay model than a perturbation approach and has different psychological implications for learning and instruction.

To the extent that students do use faulty adaptations of base rules in new situations (as described earlier), then a "repair theory" of bugs may synthesize the alternative constructs of deviations and mistaken inclusions/omissions (VanLehn, 1983). When a student reaches an impasse in solving a problem because his or her current skills are insufficient, he or she may attempt to "fix" an existing procedure to make it usable. Such an erroneous extrapolation of a correct procedure can produce results consistent with both types of "buggy" theories.

The "repair" approach to student errors generates complex predictions involving "star bugs" (which never occur in studies of human errors and therefore should be impossible within the student modeling paradigm) and "core procedures" (transmitted by instruction and used by learners as the basis from which variants and deviations are made). The short-term instabilities in student problem solving observed in protocol analysis are attributed to learner experimentation with different repairs until a satisfactory mixture of modified procedures is found. This type of student model implies that instruction should be based on core procedures structured and sequenced to minimize the chances of faulty student extrapolation/adaptation.

Sleeman (1984c) hypothesizes that student mis-generalization in reasoning may be important in causing errors. To the extent that pupils adopt problem-solving procedures by inferring generalized procedures from instructor examples, "mal-rules" may be formed. These inappropriate strategies can be difficult to diagnose. For example, many possible alternative types of mal-rules may explain a given sequence of pupil errors (a combinatorial

explosion similar to that discussed in the student knowledge section). Also, even students who have a rote mastery of correct procedures may revert to mis-generalization under conditions of cognitive stress, producing fluctuations in problem-solving behavior.

The continuing evolution of competing theories on student error has created a number of questions for ICAI research. How generalizable are the error taxonomies derived from protocol analysis of a particular domain? How can problem solving be structured to reduce the occasions on which pupils need to attempt repairs? When a learner reaches a problem solving impasse, how can his or her extrapolations of base rules be improved? How can instruction on core procedures minimize false inferences? To what extent can perturbation constructs substitute for overlay and differential models of student error?

Student Learning

Providing a general overview of learning research in motivational and developmental psychology is beyond the scope of this paper, so the focus in this section is on knowledge acquisition studies in cognitive science. (Another type of research not reviewed here, but of potential interest in developing computer tutors and coaches, is machine learning (Michalski, Carbonell, Mitchell; 1983.) The previous section suggests that understanding the causes of student error would lead to better principles of instruction. Research on the "felicity conditions" underlying learning may create such a link.

As a way of focusing attention, pupils expect the teacher to obey certain conventions in the communication process (VanLehn, 1983). For example, students anticipate that each lesson will introduce one new, simple piece of information; that this skill will add to or substitute for already learned

procedures; and that the information can be induced from examples and exercises. (This last expectation is similar to the assumptions on pupil inference and generalization in the discussion on mal-rules.)

These illustrative felicity conditions provide a model for knowledge integration--the construction of a skill from subskills--as opposed to knowledge compilation. The focus is the learning expectations governing the pattern of communication from teacher to pupil. When this process of information transfer fails, students reach an impasse in problem solving and are forced to attempt repair to their existing skills.

An alternative approach to this step-by-step model of learning from instruction is the simulation of student knowledge acquisition by production systems (Anderson, Farrell, Sauers; 1984). Computer programs have been developed which mimic the hierarchical organization of problem solving and the goal-driven cognitive control mechanisms observed in protocol analysis of pupils. In these simulations, structural analogy to concrete cases is an important learning mechanism; acquired knowledge is then compiled in ways similar to the automatization of expertise discussed earlier under knowledge representation. This model suggests that the limited capacity of human short-term memory may be a major factor in the dynamics of learning. (An overview of production system learning models is given in Klahr, Langley, Neches (1984.))

Other computationally related studies address the relationship of learning to user interface design and human factors analysis (Moran, 1982). For example, Malone (1982) describes design heuristics for creating enjoyable "learning by doing" interactions between student and ICAI device; these include use of challenge, fantasy, and curiosity. RABBIT (an intelligent database assistant) uses a retrieval paradigm based on reformulation, a

psychological theory of human remembering (Tou et al., 1982). A variety of studies on cognitive process analysis of learning are collected in Snow, Federico, Montague (1980).

Research on student learning has multiple implications for ICAI systems. For a given subject, what is the optimal set of felicity conditions to guide communication between tutor or coach and student? Beyond expectations for information transfer, what other factors are important in maximizing knowledge integration by learners? What are the implications for ICAI design of production-system simulations of student error and learning? What conclusions from human factors and user interface design can guide curriculum developers?

Conclusion

Regardless of interest in ICAI, studies in cognitive science and machine learning will continue. In general, however, research on student modeling is heavily linked to the construction of computer tutors and coaches. Other AI-based educational tools (such as "idea processors" or "enpowering environments") do not require internal representations of the learner.

As indicated above, student modeling research touches on topics at the leading edge of artificial intelligence and cognitive science. In addition, the advent of inexpensive devices for measuring a user's physiological reactions may open a new and complex area of study: the sensing of learner consciousness and mood as a component in instruction. Overall, the challenges faced in learner representation may be the most difficult in ICAI.

Teaching Module

Given expertise in the subject domain and a model of the student's present comprehension, an optimal intelligent tutor or coach would select an

efficient path through its knowledge base to generate expert behavior by the user. Initial teaching strategies, based on a prototype or on a pupil's previous performance, would be modified as the student model evolves. The pedagogical strategies used might include presenting increasingly complex concepts or problems, simulating phenomena, Socratic tutoring with correction of pupil misconceptions, and modeling of expert problem solving via coaching.

The ICAI system must have a discourse-oriented theory of explanation to coordinate these teaching strategies. This instructional theory would incorporate rules for which pedagogical means are most efficient to accomplish a given end, alternative approaches to dialogue management (adjusting to different learning styles), and domain-dependent teaching heuristics (such as those suggested earlier in the discussion on genetic graphs). Some of these skills can be derived from protocol analysis of expert teachers; others have no counterpart in human pedagogy since they utilize attributes unique to computer-based instruction (such as simulation of complex phenomena).

The responsibilities of the four ICAI modules have considerable overlap, and many issues related to teaching have already been discussed in the sections on expertise and student modeling. Making the knowledge base articulate, using simplification rules (such as the kernel models described under the nature of meaning), and sequencing subject matter involve both the expertise and the teaching sectors of an intelligent coach or tutor. Given this bridge to the knowledge base, the instructional component can infer the relative difficulty of items in the syllabus and can build interconnections to previously learned information in other domains. Similarly, felicity conditions and the debugging of the user model link the student and teaching modules, promoting individualization of instruction.

No universal approach to modularization exists in ICAI. One researcher

may locate responsibility for felicity conditions in the student model; another, in the teaching sector. This paper stresses the evolution of the intelligent coach or tutor from expertise through student modeling to teaching, with only the explanatory aspects of pedagogy located in this instructional section.

No attempt has been made to summarize the vast amounts of educational research on teaching. Such an endeavor is beyond the scope of this study, and also much of this information is irrelevent because of its focus on group rather than individual learning. Rather, the discussion below is confined to recent computational and cognitive perspectives on pedagogical processes.

Paradigms for Explanation

The fundamental issues for a tutor or coach are whether to intervene in the information flow, what to discuss, which presentation strategy to use, and how much to say. In a learning-by-doing environment, intervention takes the form of an interruption by the coach when a systematic pattern of error has been spotted. In tutoring, intervention involves judging when to shift between imparting new information and debugging the pupil's current conceptions.

One major instructional factor identified by ICAI researchers is the effect of discontinuous information flow on student interest. Too frequent interventions destroy student initiative and decrease motivation to learn. Important criteria in choosing when to interrupt are relevency and memorability; the information provided by the intervention should be directed at a particular weakness, useful in the immediate situation, and demonstrably superior to the pupil's misconception (Burton and Brown, 1982). (These issues are illustrated in the discussion of WEST in part three of this paper.)

When an interruption is indicated, the choice of what to emphasize becomes the next fundamental issue. Means-end guidance rules can be used to relate errors found in the student model to the selection of experiences which will remove those misunderstandings (O'Shea, 1982). These rules reinforce mastery by gradual accumulation of progressively more difficult knowledge. This can involve multiple types of explanation; here, genetic graphs can be a means of deriving alternative examples and illustrations closely linked to previously learned knowledge (Goldstein, 1982).

Sometimes, choice of content involves more global issues than responding to a particular pattern of student error. For example, at frequent intervals a review of what has been learned so far is a useful tutoring technique. Also, if a pupil is struggling to master a learning-by-doing experience, reducing the overall level of difficulty by simplifying the task can enhance motivation, diagnosis, and remediation.

Selecting which strategy to use in transferring knowledge from ICAI system to student is the third fundamental issue in explanatory efficiency. Knowledge of the user's learning style from the student model is an important criterion for selection, as is the choice between descriptive (textual) and depictive (graphic) representation. For example, instructional strategies with "visibility" (e.g., illustrating programming processes step-by-step via a simulated, simplified machine) are useful in teaching novices computer programming languages (du Boulay, O'Shea, Monk; 1981). Higher order cognitive skills (such as the ability to visualize what a program is doing) can be strengthened using this approach (Lieberman, 1982).

A typology of explanations is important for a computer tutor or coach in choosing among a repertoire of possible strategies (Stevens and Steinberg, 1981). For example, the STEAMER system incorporates nine types of

explanations--including physical-causal, information flow (feedback), topological (connectivity of components), and attributional (state changes)--in teaching students about navy propulsion plants. Each explanatory category has its own characteristics which dictate an appropriate instructional approach.

A theory of rhetoric (stating the explanation so that it will be understandable) is vital for ICAI. Constructing explanations to minimize complexity (through considering such aspects as focus of attention and embedded discussions) is a structural approach to rhetoric (Weiner, 1980). Alternatively, NEOMYCIN uses an abstract (domain independent) representatation of strategy to guide the choice of which tutoring rule from its repertoire is most appropriate in an explanatory situation (Hasling, Clancey, Kennels; 1983). (This is an extension of the GUIDON work covered in section three of this paper.)

The Socratic tutor has been a powerful instructional model in ICAI. The mixed initiative dialogue fostered by this teaching strategy gives an opportunity to refine the student model, pruning the tree of possible theories the pupil might hold about a particular situation. The goals of the Socratic tutor are to follow a domain-dependent script of knowledge (as discussed in part three under the WHY and SCHOLAR systems), using production rules to identify student misconceptions and to offer counterexamples which build a new understanding (Stevens and Collins, 1980).

How much to say is the fourth fundamental issue in an explanatory paradigm. In indicating pupil error, learner self-image and motivation must not be damaged. The instructional response must be at an appropriate level of detail, neither so fine-grained as to be boring nor so general that the

student is uncertain of its application to the problem at hand (Wallis and Shortliffe, 1982).

A theory of hints is important here. Burton and Brown (1982) describe four successive levels of hints in WEST which range from indicating a weakness to describing in detail how to make an already indicated optimal move. The student's response to varying levels of explanation provides a useful diagnostic mechanism for assessing how deep an understanding has been attained.

Ideally, the intelligent coach or tutor would be capable of self-improvement, expanding and refining both its subject matter knowledge and its explanatory strategies with accumulated experience. In some subject areas, an ICAI system can be designed to acquire superior problem solving approaches from its students (Kimball, 1982). Tutors based on production rules can be equipped with a component which makes experimental changes in teaching approach, adding successful modifications as they are discovered (O'Shea, 1982). However, the amount of improvement which can be gained for complex and less structured subject matter is questionable.

Multiple questions for ICAI arise from research on the nature of explanation. How much can traditional educational research on group instruction be applied to individualized computer coaches or tutors (and vice versa)? To what extent can intervention criteria such as relevency and memorability be assessed independent of subject matter and student attributes? Within what limits can typologies of explanation, theories of hints, approaches to example selection, or rhetoric strategies be generalized to other domains? What proportion of total instructional efficiency can be attained via self-improvement? These illustrative questions indicate the range of issues raised by studies of teaching strategy.

Conclusions

The teaching module is the least studied of the four ICAI components. Expertise and communication research both are proceeding independent of interest in intelligent tutors or coaches, and user emulation has attracted more interest among cognitive scientists than have explanatory processes. Also, sophisticated instructional strategies are likely to be useless unless linked to a strong base of expertise and a powerful diagnostic model, so early work has focused on these areas.

However, all the other components are useless without a teaching module to integrate and coordinate their functions. Intelligent coaches and tutors (unlike other AI-based educational tools) require a means of structuring learning-by-doing to maximize efficiency and effectiveness. Traditional educational research, with its focus on group training, has not supplied an adequate theoretical framework for optimizing individual learning in information rich environments. An increased research emphasis on explanatory strategies is vital for the success of ICAI.

Communications Module

Communications issues in ICAI largely parallel research themes in the field of natural language comprehension and generation, which is a major area within artificial intelligence. A review/synthesis of the many studies in this field is beyond the scope of this paper. Instead, this section will give a brief overview of those research topics which have special relevence to intelligent tutors and coaches.

The focus will be on linguistic work, since non-linguistic input (e.g., mice and pull-down menus), while offering convenience in user interface design, severely limits the depth of communication which can occur. In the

near future, the means by which information is exchanged between student and computer will likely continue to be non-vocal. While voice output for ICAI systems is progressing, voice input is still a major technical challenge.

A general analysis of discourse indicates the need for three types of information in carrying on a dialogue (Winograd, 1977):

--knowledge about patterns of interpretation (to understand a speaker) and action (to generate utterances) within dialogues (Faught, 1977)

--domain knowledge needed for communicating content

--knowledge of each speaker's intentions within the overall communications situation (similar to the felicity conditions discussed under the nature of learning)

These necessarily involve linking the communications module to the ICAI system's expertise, student model, and instructional strategies.

For example, GUIDON (discussed in part three of this paper) uses discourse procedures to indicate when a particular tutoring rule may be appropriate for communicating about a domain rule (Clancey, 1982). The communications module in this system has access to three levels of domain knowledge: information used in expert performance, support data which explain the justification for expert actions, and abstraction knowledge identifying patterns in the performance knowledge. The student model incorporates knowledge of the communications situation via an overlay representation of the pupil, a syllabus, and a dialogue continuity mechanism.

Beyond an analysis of discourse, the most fundamental issue in designing an ICAI communications module is language comprehension. From an artificial intelligence perspective, this can be the most difficult technical problem in a computer tutor or coach. Educational applications are particularly challenging, for two reasons: First, the nuances in a response are vital in

shaping the student model, so more than crude comprehension of a statement is required. The sophistication in analysis of utterances needed may be greater for ICAI than for any other major application of natural language comprehension. Second—unlike human instructors, who can interpret pupil behavior through multiple indicators such as visual cues and tone of voice--the ICAI system must currently rely on one dimension of input. (In a few years, hand-held physiological sensing devices may offer additional information on states of consciousness and mood.) This limited sensing capability makes the problem of assessing student understanding much more difficult.

As its approach to language comprehension, SOPHIE uses a semantic grammar to parse user input (Burton, 1976). Speed (a problem for computer-based systems) can be greatly increased by focusing on the meaning of key words in a response rather than attempting to understand the utterance syntactically. This approach aids in dealing with linguistic ambiquities such as anaphoric deletions (implied references to earlier parts of the dialogue) and ellipses (omitted words to be deduced by inference). (Part three of this paper illustrates several of these points in its discussion of SOPHIE.)

However, more than semantic and syntactic expertise are necessary for understanding a statement in natural language. Contextual knowledge of real-world phenomena and common sense about standard types of occurences are vital in comprehension. Representing this sort of knowledge in a computer is a very difficult problem in artificial intelligence and may be a significant barrier to the types of instruction that a tutor or coach can present.

Language generation, while still challenging, is a less complex issue with fewer idiosyncratic issues in educational applications. When producing utterances, planning to achieve communicative goals is a critical factor. The

TELEGRAM (TELEological GRAMmar) system encodes discourse information in a specialized formalism with linguistic knowledge and planning control structures (Appelt, 1983). If a system has reason to believe that its planned response may mislead the user, then "informing behavior" (identifying and avoiding potentially confusing utterances) is a useful strategy (Joshi and Webber, 1984).

All these types of research have implications for ICAI systems. What are typical classroom patterns of interpretation and action for dialogues? Do educational situations utilize discourse procedures different from those in most communication situtions? To what extent do lack of contextual knowledge and "common sense" limit the proportion of the curriculum which can be communicated by intelligent tutors and coaches? What are the "informing behavior" techniques used by expert teachers? These questions illustrate the range of issues which the communications module (a "service" function within the ICAI system) imposes.

CAPABILITIES AND LIMITATIONS OF CURRENT ICAI SYSTEMS

As this discussion of major research themes indicates, much work remains to be done before building intelligent tutors and coaches is well understood or routine. On the other hand, the feasibility of these devices has been demonstrated by implementations such as those described in part three of this paper, and significant progress has been made on many crucial issues. Overall, what are major illustrative strengths and weaknesses of ICAI at present?

One limit on the deployment of AI-based educational tools has slowly been disappearing: small computers are steadily becoming less expensive and more powerful. As a result, desktop machines with memory capacity and processing

speed sufficient for ICAI are beginning to emerge (current system prices are about five thousand dollars). While at present these computers are sold primarily to businesses, as prices fall and as educational software capable of using such power is developed, schools will begin to purchase these systems as the next generation of instructional tools.

The widespread dissemination via smaller computers of sophisticated ICAI implementations such as STEAMER and SOPHIE is likely to increase interest in computer coaches and tutors. Historically, the impetus to develop AI-based educational tools has come from the military and industry, where training costs are high and the consequences of low effectiveness are profound. More funding may come from other sources as other groups within society begin to realize that these devices can be useful and will become affordable.

Growing commercial interest in expert systems is also having a positive impact on ICAI. Specialized software for knowledge engineering is increasingly available, and funding is expanding for research studies on the nature of expertise. However, an expert computer problem solver requires only mechanically optimized knowledge, while the representation of expertise in a computer coach or tutor must include a "glass box" model and knowledge sequencing. Thus, research on expert systems will not automatically provide the full range of studies necessary.

Also, the realization that "in the knowledge lies the power" may reduce the potential usefulness of ICAI devices and make their production for the full range of curricular subjects a time-consuming and expensive process. The understanding gained from subject-domain experts of what reconstructed knowledge is most useful in building authentic knowledge will be valuable, but attaining this understanding will require far more complex curriculum development strategies than currently employed. If knowledge representations

must be narrow and specialized, many individual coaches and tutors will have to be created, each handling only a small portion of the curriculum.

Of course, by careful choice of topics, maximum generalization can be achieved. For example, SOPHIE's instruction on power amplifiers may provide the foundation necessary for teaching most types of sophisticated electronic troubleshooting. Nonetheless, the deployment and integration of numerous specialized tools is a less ambitious goal than the more generic instructional devices thought feasible in the early years of artificial intelligence research.

As with knowledge representation, the research themes vital for the communications module of an intelligent tutor or coach are of great interest to the general field of computer science. Extensive commercial possibilities for devices capable of language comprehension and generation also ensure continuing research in this area. However, progress in linguistic communication by computers has been slow, and the limited user interface characteristic of current ICAI systems may be the most difficult technical hurdle to overcome.

These problems may restrict the range of subjects for which computer tutors and coaches can be developed (since some types of content demand sophisticated comprehension of student utterances by the instructor). In addition, a limited communications format constrains educational effectiveness by reducing user motivation and limiting the inferences which the student model can make. As perhaps the most demanding application of language comprehension and generation, the evolution of ICAI is dependent on continued progress in this area.

The explicitness required in all modules of a computer coach or tutor is a mixed blessing for educators. Having to specify operationally every detail

forces a deeper comprehension of subject matter, learning, teaching, and communication; this enriches the human instructional enterprise independent of the development of ICAI devices. New models and metaphors, novel ways of conceptualizing familiar problems, and a more profound understanding of the control variables in educational effectiveness are emerging from studies of AI-based tools.

However, the requirement of explicitness means that only an intellectually rigorous production effort can develop effective computer coaches and tutors. Every aspect of each module must be close to perfect if a worthwhile educational product is to result. Hastily executed applications will be of questionable value, and prototypes will require extensive testing and debugging before dissemination. This "all or nothing" characteristic is a distressing attribute of ICAI, as intermediate stages of development will not be directly useful in improving instruction (although indirect benefits from knowledge that improves human teaching will be important).

Because traditional educational research has focused on group instruction, little is known about some individual learning characteristics vital in developing the student and pedagogical models. The long lead times from conceptualizing a study to disseminating its results mean that research begun today will reach fruition about the time that computers sufficiently powerful for ICAI are affordable by schools. However, few investigators in education have shifted the focus of their work from current concerns to this emerging new paradigm for instruction, despite attempts by cognitive and computer scientists to promote such efforts.

Unfortunately, researchers in artificial intelligence and cognitive science have often not followed their own precepts on expertise in studying student and pedagogical representations. Rather than using knowledge

engineering to extract authentic knowledge for veridical situations from human teachers, many ICAI studies build elaborate speculative edifices on theories derived from small-scale, ad hoc observations. The high cost of better designed research and computer scientists' lack of familiarity with pre-college instructional settings contribute to this problem.

Difficulties due to inadequate empirical research will intensify as the focus of investigation shifts to earlier developmental levels where generalizations from college and industry populations are less likely to apply. For example, some pupils may have a problem-solving paradigm which is so divergent from an expert strategy that overlay, differential, and perturbation student models are all inadequate. Large-scale studies to indicate the number and type of fundamental mindsets individual learners bring to a given subject would be of great value in producing computer coaches and tutors, yet little research is occuring in this area. Both a refocusing of instructional investigation on emerging issues and increased collaboration among educational researchers, artificial intelligence specialists, and cognitive scientists would help to improve the quality of current ICAI development.

Overall, increasing availability, decreasing cost, and growing commercial interest in AI-based tools are enhancing the potential of ICAI. Limits on the sophistication of communications modules, on the scope of subject domains, and on current understanding of individual learning all are constraining the effectiveness of computer coaches and tutors. The explicitness required for constructing intelligent devices makes progress more difficult and time consuming, but enriches the theoretical perspective which emerges.

Conclusion

Early researchers in AI were very optimistic about how quickly computers would be competitive with humans in many aspects of intelligence. Over the past two decades, some major goals (such as the General Problemsolver) have been abandoned as infeasible; others once thought easy are now assessed as very difficult (e.g., machine translation). Thus, some degree of scepticism about claims of AI rapidly revolutionizing educational practice seems appropriate.

However, the capabilities discussed so far in this study are conservative claims of performance, and progress in these areas of AI research has been steady, if slow. Moreover, economic constraints on the effectiveness of the traditional model of instruction are gradually eroding quality. A cost-effective pedagogical approach based on teacher/tool partnerships (see section five) may be seen as an increasingly attractive alternative to spiraling costs. What are the potential long range changes in schooling which may result from ICAI?

SECTION V: THE POTENTIAL OF INTELLIGENT COMPUTER ASSISTED INSTRUCTION FOR EDUCATION

We find ourselves divided on the question of ICAI's potential contribution to education—divided individually rather than severally. That is, each of us is of two minds. On the pessimistic side, several factors suggest that the near-term contribution may be limited or even negligible:

— Design, theory, and aspiration have greatly outdistanced concrete achievement in the field. The definition of an ICAI system in terms of four modules (expertise, student modeling, teaching, and communication) frequently turns out to be more theory than actuality. Many systems reported in the literature are only partially developed; one or two components are of principal interest to the developers, and the others are left in the conceptual or rudimentary prototype stage. This is understandable--perhaps even necessary--when the system is being built for research purposes, but does suggest that surmounting difficulties involved in creating complete systems will demand an even greater investment of dollars, time, and talent than many existing systems have received.

— Existing systems are almost completely domain-specific. There have been some efforts to abstract generalizable shells from a few of them, and these efforts should certainly be expanded, but at this point developing a system in one domain makes at best an indirect contribution to the development of additional systems.

— Because ICAI systems are both expensive and domain-specific, the costs of creating a set of systems spanning the entire curriculum in the major academic subjects appears prohibitive.

— Technical obstacles also stand in the way of broad application. ICAI seems most applicable to subject matter domains where knowledge is well-defined and of no more than moderate complexity. In areas where ambiguities and subtleties abound, the challenge appears

exponentially more difficult. As suggested by the definition between "education" and "training" below, this appears to rule out developing systems to teach, stand-alone, a great deal of the K-12 curriculum.

— The natural language problem has proved daunting, a far greater challenge than many AI experts had guessed. One ICAI researcher told us flatly, "natural language is dead," and argued that a search for alternative interface modes--especially on the input side--is essential if ICAI is ever to be practical. Others believe that natural language research is on the verge of a breakthrough. None of the present authors is an expert in the area, but we incline to skepticism on this point. Where sensitivity to nuance is not required and the scope of communication is narrow, natural language interfaces may prove feasible in cost and technical terms, but the investments required for nuance-sensitive and broad-gauge communication seem likely to prove prohibitive for the K-12 education system in the near future.

-- Beyond all of the difficulties entailed in the development of ICAI, great difficulties of implementation present themselves. The history of this century's efforts to improve education through technology is not encouraging. In fact, our experience with film, audiotape, instructional television, and language laboratories--not to mention the experience to date with computers—is rather dispiriting. Because of its potential for direct substitution of teacher activities, ICAI would seem to present a far greater implementation challenge than any of these.

In light of these considerations, some may find it a tribute to our capacity for hope rather than a result of realistic analysis that we persist in believing that ICAI does have great potential to improve education. Nevertheless, we are convinced that the method of study demanded by ICAI—careful, deep, domain-specific analysis of subject matter and of children's errors and advances in learning it, all tied to precisely specified teaching interventions--has the potential to transform research on teaching

and learning, to make educational improvement both more scientific and more productive than it has been to date.

Further, we believe that none of the obstacles and drawbacks enumerated above need prove fatal to the development and implementation of powerful ICAI systems in the K-12 education arena--if certain conditions are met. First, topics to be addressed through ICAI will have to be selected judiciously. Not only must they be in subject areas where knowledge is well-defined and limited in complexity, but they must also be chosen strategically to help students over obstacles that stump or discourage large numbers of them so that ICAI systems justify their cost be sharply increasing productivity. Second, for the foreseeable future, ICAI systems will have to make limited use of natural language, relying instead on alternative flexible, user-friendly interfaces. Third, increased attention will have to be given to approaches and techniques that permit generalization from one domain to neighboring domains and other means of reducing development costs. Fourth and most important, a large scale, national research program will have to be mounted.

If these conditions are met, what kinds of benefits might we expect from ICAI? To answer this question, we have created an extremely optimistic 15-year scenario. Such a chronological horizon provides a context within which three- to eight-year strategic plans can be evaluated. A high degree of success in developing ICAI is hypothesized: that is, a dedicated national research program, the emergence of a collective school market, and major societal requirements for adult retraining combine to produce a rapid evolution of AI-based instructional devices.

In this scenario, by the year 2000 the types of teaching functions available via information tools at a cost comparable to human instruction include:

ICAI

—mixed initiative, dialogue based tutors

—complex, interactive games and simulations with embedded coaches

Other AI

—"microworlds" (limited alternative realities) with domain dependent languages to facilitate exploration (e.g., LOGO)

—"idea processors" which allow the interconnection of concepts in an elaborate network (van Dam, 1984)

—sophisticated "empowering environments" for artistic, musical, and literary expression (Brown, 1983)

—surrogate travel and experience via interactive telecommunications and videodisc

—powerful, linked data access and management devices

These technologies might include such capabilities as voice input, monitoring the user's state of consciousness through physiological sensors, and screen control via eye movement trackers. Some as yet unconceptualized teaching functions may also be available.

This forecast assumes steady progress in both the computational and cognitive research necessary to enable devices of this sophistication. Over the next fifteen years, the rapid evolution of information technology hardware seems certain to continue (although some significant technical limits may be reached by the late 1900s). The processing power of current computers will increase by at least two orders of magnitude at constant cost. External memory will be very cheap; internal memory may involve significant expense, but will be much less costly than today.

Large, high resolution flat monitors will be reasonably priced for educational purposes. Delivering huge amounts of information over distance will be inexpensive, and high speed, high quality laser printers will be

readily available. The universality of digital code will allow small devices which combine the attributes of the telephone, television, videodisc, computer, and xerox machine. Software tools which allow the construction of very complex courseware will be routinely used.

In the next fifteen years, the availability of the knowledge about cognition needed for constructing sophisticated ICAI devices is less certain. Increasing our understanding of domain-independent thinking skills (i.e., problem solving, information retrieval), metacognition (e.g., learning from errors), and conceptual restructuring (paradigm shifts) will require time, a critical mass of research expertise, and sustained funding. Hardware and software development will be driven by forces external to education, but pedagogical and psychological research are more dependent on the emergence of both a national mandate for strong universal education and an economic incentive for meeting this need via information technology.

Societal pressures will be influential in determining not only the resources available for developing ICAI technology, but also the purposes for which these devices are used. In general, the goals and structure of schools are strongly shaped by external forces; Sputnik, the Civil Rights movement, and recent economic malaise all have caused more educational change than the past century of internal innovations. Over the next decade, a cultural conventional wisdom on how to use the new information technologies will emerge. Whether the predominant pattern is "automation" or "person/tool partnerships" will be very influential in shaping the evolution of ICAI.

Some see the information technologies as best suited to producing robots, dedicated intelligent machines, and other artifacts which can work without human operators. Computer tutors and coaches fit into this category of "automation" (substitution of technology for people to gain efficiency and

effectiveness). If applied to the economy overall, this approach yields "superindustrialization."

In such an extreme future, national prosperity would continue to depend on heavy standardized production industries (such as steel and automobiles. At present, American companies are at a disadvantage in the international arena because of rising labor costs, natural resource depletion, expensive energy, high interest rates, and technological obsolescence. Through automation based on very sophisticated information technologies, the U.S. could attempt to increase productivity enough to become dominant again in these traditional markets.

Researchers have analyzed what such an economic development policy might mean for education. The occupational mix in a superindustiralized society would have a small percentage of scientists, engineers, and policy setters; few middle level professions; many low level waiters, janitors, and service jobs; and large numbers of unemployed sustained by income redistribution.

Governance would shift toward technocracy (decision making by expertise) rather than democracy. The major purposes of ICAI in this future might be to screen for small numbers of talented elite while training the vast majority of students for a menial social status; other AI-based devices would aid in preparing those few workers needing advanced cognitive skills.

An opposite extreme future would involve America's making a transition to a knowledge-based economy (moving beyond automation of industrial processes to value-added crafting of information). Each item produced for the global marketplace would be customized to the needs of its owner rather than mass-produced to some average set of specifications (the equivalent of having all one's clothes tailored). Such a strategy for economic evolution would require the use of information technologies predominantly in "person/tool

partnerships" (a human operator and an information device together accomplishing more than either could alone). Some unemployment might occur during the transition to this knowledge-based economy, but long term many high skill jobs would be available. Governance would shift toward more decentralized forms of decision making as information networks lessened the need for elaborate systems of representing each citizen's interests.

Analysts have studied what the implications of such developments might be for education. To provide the creativity and flexibility necessary for custom design of products, workers would need higher order cognitive skills complementary to the strengths of a sophisticated information tool. As its portion of the partnership, the technology would accomplish most types of standardized problem solving, while human operators would be responsible primarily for problem recognition and responding to unusual situations.

As the information tools become capable of many functions now taught as the foundation of vocational advancement, creativity, flexibility, decision making given incomplete data, complex pattern recognition, information evaluation/synthesis, and holistic thinking would become central occupational skills. A new definition of human intelligence would emerge, based on what the new technologies lack.

If people were hired on the basis of these higher order cognitive characteristics, new types of authority structures would evolve in response, moving beyond "following orders" to take advantage of workers' expanded attributes. A trans-hierarchical approach to decision making could decentralize power and responsibility while retaining speed and accountability. Cooperation, compromise, and group decision making abilities would become important universal skills.

With the power of the technology continuing to double every two years for

the same cost, tools would be frequently redesigned, leading to the need for massive adult retraining as occupational skills rapidly shift. The whole range of AI-based devices would be universal educational aids in such a future, since all students would need the sophisticated knowledge these tools can impart. Funding for research in cognition to enable the development of complex instructional devices would be more likely in this future than in the superindustrialized scenario.

Thus, the predominant cultural pattern of technological usage America chooses (automation versus person/tool partnerships) will strongly shape the goals of schooling. What may these large scale changes mean for individual teachers, students, and administrators? How may the process of classroom learning alter?

Implications For Teachers

As ICAI devices evolve, the nature of education will gradually shift. Historically, the pendulum of pedagogical innovation has swung between extremes of "structured instruction" (teacher as source of all knowledge) and "unstructured learning" (student obtains information via trial and error experience). To maximize the effectiveness of the new instructional technologies, much of the curriculum may be taught through "structured learning." In this approach, the pupil discovers knowledge within an organized, information-rich environment. The challenges faced are sequenced and tailored to individual needs, with help from teacher or technological device available as required.

The traditional model of instruction is oriented to groups; the teacher spends most of his/her time speaking to the needs of the middle sixty percent while trying to keep the top and bottom students peripherally involved. The

efficiency of this approach is low for any material which involves mastering a standardized problem-solving technique, as each of twenty or more pupils may be at a different point in learning the algorithm. As a result, skills such as division, which could be acquired in a month by a developmentally ready learner working one-on-one with a skilled teacher, can take years of classroom practice to achieve.

Group instruction may be the best way to teach some types of higher order thinking and communication skills, where interaction and pooling of ideas are important. Even in this situation, however, optimal class size varies by type of material, and students may have widely differing masteries of the basic skills which are a necessary foundation for complex thought. This greatly reduces educational productivity and increases disparities in pupil performance.

Unstructured learning situations such as discovery learning and apprenticeships, while alternatives to the traditional model, have historically also had efficiency problems. Students need a series of exploratory environments which offer gradually increasing difficulties, as well as external intervention to aid in conceptual restructuring when a dead end is reached (Brown, 1984). Many of the post-Sputnik science and math curriculum reforms failed because neither pupils nor teachers were well prepared to filter knowledge from unstructured environments heavily loaded with information.

Thus, classroom instruction has been hampered by the teacher's powerlessness both to tailor the size of the group to the content being learned and to intervene on an individual basis as needed. ICAI devices offer a way of changing this situation, since for some portions of the curriculum these could provide stand-alone individual and small group instruction. For

all material involving a limited range of right answers, mixed initiative tutors and coaches could guide pupils to the right problem-solving technique, recognizing errors, providing remediation, and modeling expert performance as requisite.

Imagine dividing the curriculum into two parts: "training" (limited range of right answers) and "education" (multiple right answers, answer unknown, human relations). ICAI devices cannot, stand-alone, provide "education" (although AI-based tools would be valuable supplements to the teacher); they cannot recognize the full range of right answers or provide human interaction. Technological tutors and coaches could substitute for people in "training" students, however, freeing the teacher to work with pupils flexible and giving learners a more uniform preparation in basic thinking skills. This in turn would increase instructional productivity in group situations, so both training and education would become more efficient.

ICAI also increases quality in several types of instruction beyond even what a skilled teacher could offer. A human instructor cannot simulate a device or a game situation or an alternative reality or surrogate travel as convincingly as can information devices. Nor can a single person become expert in all the types of abstruse knowledge from which gifted students might profit. Also, symbolic manipulation (whether in graphic, literary, musical, or numeric form) is a strength of computers; expecting teachers to compete is like pitting John Henry against the steam engine. For many parts of the curriculum, a person will always be superior to a machine, but--even if costs were equal--in some situations AI-based devices are optimal.

Indirectly, the information technologies will change what teachers must know about their subject matter. "Basics" (such as arithmetic and algebraic manipulation in math or spelling in composition) will continue to be taught,

but only to the mastery level required as a foundation for higher order thinking and usage. Information tools such as calculators and spelling checkers will provide the performance proficiency now required of people. In general, descriptive and declarative (what) knowledge will be deemphasized relative to procedural (how) knowledge. Information technologies require this type of human expertise to maximize their usefulness, and ICAI gives a means of imparting process skills efficiently.

In addition to mastering a different level of knowledge about their subject, instructors will require an expanded range of teaching techniques to take full advantage of partnerships with tools. Human skills need to complement ICAI devices, accomplishing what they cannot. Teachers, regardless of subject area or grade level, must acquire the following attributes if the benefits of instructional technology are to be maximized:

--skills in facilitating student learning through individual and group interaction in information-rich environments, possibly including multilevel age groups and flexible time allocations (Dede and Adams, 1984)

—an understanding of what stand-alone instructional tools can and cannot accomplish ("training" vs. "education"); a sense of when an information device may be superior to human pedagogy

--greater procedural knowledge of the subject area; an ability to tailor the size of the instructional group to the type of content being communicated

--higher order cognitive skills such as creativity, flexibility, decision making given incomplete data, complex pattern recognition, information evaluation/synthesis, and holistic thinking (the new definition of intelligence)

—the ability to use sophisticated diagnostic and assessment techniques to evaluate student attainment of complex cognition

Skill in operating and maintaining the new instruction tools is only a small part of this list. The indirect outcomes of ICAI (empowering a new role for

teachers) are greater than the direct effects (substituting for some types of instruction).

Beyond these implications of ICAI, the use of information technologies in general will alter the role of teachers in several ways. Much of the boring, repetitive, time-consuming drudgery of record keeping will disappear as sophisticated, linked data access and management tools become routine in school districts. The teacher's role in promoting equal educational opportunity will increasingly include tailoring instructional devices and courseware to special needs. When a pupil spends significant amounts of time working alone with an information tool, the teacher may intensify that student's interpersonal interaction the remainder of the day to compensate for affective experience lost (Dede and Gottlieb, 1985).

Should the predominant cultural pattern of information technology use evolve into person/tool partnerships, the purpose of instruction would shift. The major goals of schooling would become:

--on the cognitive level, developing in each student the new definition of intelligence discussed earlier

--on the affective level, building skills in cooperation, compromise, and group decision making (modeling in classrooms the trans-hierarchical authority structures emerging in the workplace)

--on the normative level, socializing pupils to the complex citizenship roles of a knowledge-based society

More sophisticated assessment techniques than the multiple choice, paper and pencil methods presently used would be needed to monitor progress toward these goals.

Teacher skills in flexibility, creativity, and decision making given incomplete data would lead to shared allocations of power and responsibility with school administrators. Interactions with parents and community would increase via commercial telecommunications networks interconnected throughout

neighborhoods. Above the primary level, a significant amount of instructor time might be spent retraining adults for worker/tool partnerships as part of the societal economic transition.

Overall, whether or not America moves to a knowledge-based society, the impacts on teaching of AI-based devices will be profound. To accomplish the changes discussed above, constant inservice development of instructors will be needed, since pedagogical skills will become increasingly more complex as the information tools continue to evolve. Differential compensation based on more specialized teacher roles and skills will likely be implemented, necessitating a shift in current union approaches to occupational enhancement. A different type of person will be needed (and attracted to) teaching; pay competitive with other demanding intellectual jobs, respect from the society, and better working conditions will be essential in recruiting such people to the profession. (Dede, 1981).

If all this instructional innovation were to occur, the gains in educational productivity and effectiveness from ICAI and AI-based tools could be very significant. "Training" portions of the curriculum might be accomplished in one-third or less of the time now required. "Education" of students would be approximately twice as efficient. However, unless the total time pupils spend in school were diminished in response to this greater efficiency, overall costs of instruction would likely increase, because of the capital investment required to enable ICAI development and the enhancement of the teaching profession needed to maximize its usefulness.

The degree to which such a shift in the traditional educational model occurs will be determined primarily by how much citizens believe that universal high quality education is an essential long term investment. This

in turn will depend on whether the U.S. moves toward a superindustrialized or a knowledge-based economy.

Implications for Learners

In structured instruction, with the teacher the focal point of the classroom, students have little input into their education. Critical decisions about content, sequence, time, and priority have already been made. Pupils have few opportunities for learning by doing, asking questions, inventing ideas, linking formal and experiential knowledge, doing research, or making decisions.

ICAI devices offer an opportunity to change this situation without the inefficiencies of completely unstructured learning. Mixed initiative, dialogue-based tutors would give pupils the chance to interact one-on-one, simultaneously learning the subject and mastering skills in questioning and researching. Interactive games and simulations with embedded coaches build the student's experience, linking theory with practice and decision making. To the extent that ICAI is designed for small group usage, skills in cooperation, compromise, group decision making, and communication are enhanced.

The learner's having the chance to play a more active role in shaping his or her education is fundamental both to motivation and to eventual responsible functioning as an adult. Also, the efficiency in learning which ICAI provides could enhance the self-image of many students whose idiosyncratic needs retard subject mastery in traditional group settings. Moreover, procedural knowledge and the new definition of intelligence can best be acquired through customized, hands-on experiences which teachers without AI-based tools would be hard pressed to offer.

The relative independence of ICAI from the teacher and emerging

telecommunications links among home, school, workplace, and community offer opportunities to tailor education to the developmental needs of students. Young children are capable of far more learning than schools and families currently promote. Structured education from birth by a combination of people and information tools could unlock a vast amount of human potential difficult to achieve with later intervention. To capitalize on this developmental "window," families and teachers could work in close collaboration, using ICAI devices to decentralize and coordinate the delivery of instructional services.

Also, in a similar manner students at more advanced developmental stages could spend considerable time outside of school in community and workplace environments. During the years around puberty, for example, children have little interest in formal academic subjects compared to their curiosity about practical life skills and self-knowledge. Older pupils could begin to clarify vocational goals by experiencing work alternatives through part-time apprenticeships. Such learning activities would both contribute to society and reduce the number of teachers needed at those grade levels (allowing an intensification of human resources in early childhood, when this is most needed).

Of course, stand-alone instructional devices could provide only a small fraction of a student's education. Children need concrete experiences to build their thinking skills, and many youngsters are capable of only limited motivation and concentration apart from adult supervision. Some pupils, too, will have learning styles or special needs which respond only to human instruction. For all students, emotional development and personal growth necessitate large amounts of affective interaction without technological intermediation. Moreover, while "training" can be done with ICAI largely independent of a teacher, "education" cannot.

Overall, however, AI-based devices have enormous potential to empower learners. If all the innovations discussed above were implemented, graduates of secondary school would be more adept than most present adults in higher order thinking, emotional maturity, working with others, decision making, accepting responsibility, citizenship, practical life skills, realization of self-potential, and dealing with uncertainty. ICAI is necessary to the realization of such a future, but is certainly not sufficient.

Achievement in academic subjects by average students at the end of high school might increase to the equivalent of a senior in college, and all pupils would attain close to their individual developmental potential. Some of these results would be direct outcomes of the skills ICAI can convey; most would occur only if the overall educational flexibility that information technology could create were realized.

Implications for Administrators

The adoption of AI-based devices in schools would require continuous inservice development of teachers in response to the evolving attributes of information tools. Many educational changes have failed because instructors were not prepared for their usage and no incentive was given for innovation. Especially with demographic and economic forces creating an aging teacher population in many schools, persuading staff to shift from decades of using the traditional classroom approach could be very difficult. Teachers are likely to be particularly threatened by ICAI, which can substitute for human instructors in some applications.

Therefore, administrators have a number of leadership challenges in facilitating such a transition. If the evolution of a new instructional model is to be successful, institutional reward systems and budget allocations must

be gradually reworked to promote professional development and innovation. Retirement policies may be a means for balancing the mix of older and younger staff. Teachers will respond better to using a technological approach if the goals of education shift toward the new definition of intelligence, which requires intensive human instruction as well as the use of ICAI. Also, to the extent that schools adopt trans-hierarchical authority structures with shared power and responsibility for decisions, teachers will feel less threatened by uncontrolled change.

Implementation of all these strategies will require administrators adept at leadership as well as management. One potential technological aid in day-to-day operations is the knowledge-based (expert) system, one of the first commercial applications of artificial intelligence. Computers can give advice comparable in quality to internationally recognized experts, but this skill is limited to narrow, well understood domains where common sense, emotions, and human experience are not important factors.

Having electronic guidance available when problems arise in specialized areas of budgeting, organizational design, or facilities usage may be very useful for administrators, especially when these tools are coupled with sophisticated data access devices. Educators protentially are a large enough market that such knowledge-based systems will likely be available in the next decade. However, managers will need to be skilled in interpreting the advice given and deciding what to do. Schools would not prosper if run by a committee of electronic experts; human judgement is essential in leading any service organization.

Just as ICAI can substitute for some types of teaching, "intelligent" data management devices could eliminate a substantial proportion of the traditional middle management role in education. Part of the administrator's

function historically has been the collection and compilation of information to be passed on to higher level decision makers. Emerging technologies are increasingly becoming able to accumulate and aggregate data largely independent of human direction. This allows top policy setters direct access to information about every detail of the organization; developing ways to synthesize this enormous amount of data into "control variables" which improve the effectiveness of schools will be an important challenge.

Thus, administrator/tool partnerships will gradually change the nature of educational management. Sophisticated practices comparable to those in other service industries will become routine. As paperwork and data gathering diminish because of automated record keeping systems, the administrator will become more active in instructional leadership and in personnel management/ assessment. This will require new skills and a different career lattice, as well as continual inservice training comparable in scope to that of teachers. (Dede, 1984).

Implications for Researchers and Developers

Computers are altering the nature of educational research by changing methods of data collection, models of cognition, experimental methodologies, and approaches to assessment. Instructional devices can unobtrusively and cheaply measure key variables such as student time on task, response time to questions, percentage of errors, patterns of mistakes, and learning sequences chosen. As a result, previously unobtainable information about group performance can be garnered as a byproduct of instruction without elaborate human recordkeeping. Also, inexpensive, decentralized collection of student data in home settings can add to diagnostic knowledge (Dede and Gottlieb, 1985).

Computers enable more powerful experimental methodologies, as variables in an instructional situation can be altered one by one under the exact control of the investigator. The sophisticated training imparted by computer coaches and tutors will require evaluation by more complex forms of assessment than the multiple choice, paper and pencil tests currently used. Research into higher order cognitive skills will become increasingly important; and new models of thought, learning, and teaching emerging from artificial intelligence may initiate a scientific revolution in instructional theory. From all these changes, an applied educational science based on cognitive psychology, artificial intelligence research, ergonomics (studies of human/machine partnerships), and pedagogical theory could emerge.

The nature of curriculum development will also shift as ICAI becomes a major educational tool. Producing a quality courseware package for computer tutors and coaches will cost millions of dollars; a prohibitively expensive figure unless spread over a large population of students (in which case instruction can be delivered very cheaply). Thus, curriculum development for AI-based tools will occur at the regional or national level rather than locally, and millions of pupils may learn from the same stand-alone materials.

Conclusion

This description of potential chances only scratches the surface of the transformation which may occur. The usage of AI-based educational tools will also affect the roles of teacher trainers, instructional agents external to the school, and educational policysetters. Moreover, demographic, technological, political, cultural, and economic shifts unrelated to the new information tools will mold the ways ICAI and AI-based devices are used (Dede, 1983).

For example, the emerging biotechnologies will alter occupations and will pose difficult ethical challenges for which citizens must be educated. The erosion of the nuclear family may continue, with alternative interpersonal networks emerging to take its place; this will affect the needs pupils bring to classrooms. Immigration and migration will change the nature and size of student populations, and the mix of age groups needing instructional services will alter, shifting the major clientele for which ICAI devices will be developed.

Continuing national economic woes may place pressure on the traditional model of schooling, forcing extensive adoption of innovations which reduce costs. Societal instability and change will make planning more difficult and may affect the willingness of citizens to make long range investments in education. Over the next fifteen years, all these forces will interact in determining the evolution of artificial intelligence applications in schools.

Within a generation, the new information technologies seem likely to reshape many aspects of our lives. Whether this change will be positive or negative depends on the choices we make now. AI researchers are using computers to study how both people and information tools can gain greater control over their environment by intelligent actions. Because humans and computers have different cognitive strengths, a partnership between the two may be the best way to navigate the challenges of the next several decades and reach a bright future.

What happens as intelligent tools are infused into education will be vital in determining the quality of life fifty years hence. If ICAI is used to its full potential, the schools will develop a generation of adults well equipped to direct a knowledge-based society. If information tools are misused--or unused--the outcome is less likely to be positive. Thus, by

refocusing their efforts to exploring this new model for instruction, educational researchers can play a central role in shaping what is to come.

BIBLIOGRAPHY

Appelt, D. E. 1983. TELEGRAM: A grammar formalism for language planning. Proceedings of the International Joint Conference on Artificial Intelligence, 595-599.

Anderson, J. R. 1983. The architecture of cognition. Cambridge, Mass.: Harvard University Press.

Anderson, J.R., Boyle, C.F., Farrell, R., and Reiser, B.J. 1984. Cognitive principles in the design of computer tutors. Advanced Computer Tutoring Project. Pittsburgh: Carnegie-Mellon University.

Anderson, J.R., Farrell, R., Sauers, R. 1984. Learning to program in LISP. Cognitive Science, 8: 87-129.

Barr, A. and Feigenbaum, E.A., eds. 1981, 1982. The handbook of artificial intelligence, vols. 1 and 2. Los Altos, Ca.: William Kaufmann.

Blaine, L. and Smith, R.L. 1977. Intelligent CAI: The role of the curriculum in suggesting computational models of reasoning. Proceedings of the Association for Computing Machinery, 241-246.

Borning, A. 1981. The programming language aspects of ThingLab, a constraint-oriented simulation laboratory. ACM Transactions On Programming Languages and Systems, 3(4): 353-387.

Boyle, C.F., and Anderson, J.R. 1984. Acquisition and automated instruction of geometry proof skills. Advanced Computer Tutoring Project, Pittsburgh: Carnegie-Mellon University.

Burton, R.R. and Brown, J.S. 1982. An investigation of computer coaching for informal learning activities. In Intelligent Tutoring Systems, ed. D. Sleeman and J.S. Brown. London: Academic Press.

Brown, J.S., Burton, R.R., and de Kleer, J. 1982. Pedagogical, natural language, and knowledge engineering techniques in SOPHIE I, II, and III. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Brown, J.S., Moran, T.P., and Williams, M.D. 1982. The Semantics of procedures: A cognitive basis for training procedural skills for complex system maintenance. Cognitive and Instructional Sciences Working Paper, Xerox Palo Alto Research Center.

Brown, J.S. 1983. Process versus product: A perspective on tools for communal and informal electronic learning. Education in the Electronic Age. New York: The Learning Lab, WNET.

Brown, J. S. 1984. Learning-by-doing revisited for electronic environments. In The Future of Electronic Learning. ed. M.A. White. Hillsdale, N.J.: Lawrence Erlbaum.

Burton, R.R. 1976. Semantic grammar: An engineering technique for constructing natural language understanding systems. Report No. 3453, Bolt Beranek and Newman, Inc., Cambridge, Mass.

Burton, R.R. 1982. Diagnosing bugs in a simple procedural skill. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Carr, B. and Goldstein, I.P. 1977. Overlays: A theory of modeling for computer aided instruction. Artificial Intelligence Memo 406, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Clancey, W.J. 1982. Tutoring rules for guiding a case method dialogue. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Clancey, W.J. 1983. The epistemology of a rule-based expert system - A framework for explanation. Artificial Intelligence 20: 215-251.

Clancey, W.J. 1984. Classsification problem solving. Proceedings of the National Conference on Artificial Intelligence, Austin, Texas, 49-55.

Collins, A. et al. 1975. Reasoning from incomplete knowledge. In Representation and Understanding, ed. D. Bobrow, and A. Collins. London: Academic Press.

Collins, A., Brown, J.S., and Larkin, K. 1979. Inference in text understanding. Theoretical Issues in Reading Comprehension. Hillsdale, N.J.: Lawrence Erlbaum.

Dede, C.J. 1981. Educational, social, and ethical implications of technological innovation. Programmed Learning and Instructional Technology. 18(4): 204-213.

Dede, C.J. 1983. The evolution of the content of general education over the next two decades. Paris: UNESCO.

Dede, C.J. 1984. New technologies and education. In Encyclopedia of Educational Media Communications Technology. 2nd ed., ed. Unwin and Derick. Westport, Conn.: Greenwood Press.

Dede, C.J. and Adams, A.A. 1984. Looking into the future. PTA Today 9(7): 4-7.

Dede, C. and Gottlieb, D. 1985. The long term influence of home microcomputers on family/school relationships. Futurics 9(1): 10-18.

de Kleer, J. and Brown, J.S. 1983. The origin, form, and logic of qualitative physical laws. Proceedings of the International Joint Conference on Artificial Intelligence, 1158-1169.

diSessa, A.A. 1982. Unlearning Aristotelian physics: A study of knowledge-based learning. Cognitive Science 6: 37-75.

Douglas, S.A. and Moran, T.P. 1983. Learning operator semantics by analogy. Proceedings of the National Conference on Artificial Intelligence, Washington, D.C., 100-103.

du Boulay, B., O'Shea, T., Monk, J. 1981. The black box inside the glass box: presenting computing concepts to novices. International Journal of Man-Machine Studies, 14: 237-249.

Faught, W.S. 1977. Motivation and intensionality in a computer simulation model. AIM-305, Stanford University Department of Computer Science.

Forbus, K.D. 1983. Qualitative process theory. A.I. Memo 664-A. Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Genesereth, M.R. 1982. The role of plans in intelligent teaching systems. In Intelligent Tutoring Systems, ed. D. Sleeman and J.S. Brown. London: Academic Press.

Goldstein, I. 1976. The computer as coach: An athletic paradigm for intellectual education. AI Memo 389, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Goldstein, I.P. 1982. The genetic graph: A representation for the evolution of procedural knowledge. In Intelligent Tutoring Systems, ed. D. Sleeman and J.S. Brown, London: Academic Press.

Graham, N. 1979. Artificial intelligence. Blue Ridge Summit, Pa: Tab Books.

Hasling, D., Clancey, W., Rennels, G. 1983. Strategic explanations for a diagnostic consultation system. Heuristic Programming Project Report HPP-83-41, Stanford University Department of Computer Science.

Johnson, P.E. et al. 1982. Multimethod study of clinical judgement. Organizational Behavior and Human Performance, 30: 201-230.

Johnson, P.E. 1984. The expert mind: A new challenge for the information scientist. In Beyond Productivity: Information Systems Development for Organizational Effectiveness, ed. T. Bemelmans. New York: Elsevier.

Joshi, A. and Webber, B. 1984. Living up to expectations: Computing expert responses. Proceedings of the National Conference on Artificial Intelligence, Austin, Texas, 169-175.

Kimball, R. 1982. A self-improving tutor for symbolic integration. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown, London: Academic Press.

Klahr, D., Langley, P., Neches, D., eds. 1984. Self-modifying production system models of learning and development. Cambridge, Mass.: MIT/Branford Books.

Kuipers, B. 1982. Commonsense reasoning about causality: Deriving behavior from structure. Tufts University Working Papers in Cognitive Science No. 18.

Larkin, Jill et al. 1980. Expert and novice performance in solving physics problems. Science 208 (20): 1335-1342.

Lieberman, H. 1982. Seeing what your programs are up to. A.I. memo 656, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Malone, T. W. 1982. Heuristics for designing enjoyable user interfaces: Lessons from computer games. Proceedings of the Association for Computing Machinery Conference on Human Factors in Computer Systems, 63-68.

Matz, M. 1982. Towards a process model for high school algebra errors. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Michalski, R., Carbonell, J., Mitchell, T., eds. 1983. Machine learning. Tioga Press.

Miller, M.L. 1982. A structured planning and debugging environment for elementary programming. In Intelligent tutoring systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Moran, T.P. 1982. Eight short papers in user psychology. Cognitive and Instructional Sciences CIS-17, Xerox Palo Alto Research Center.

O'Shea, T. 1982. A self-improving quadratic tutor. In Intelligent tutoring systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

O'Shea, T. and Self, J. 1983. Learning and teaching with computers: Artificial intelligence in education. Englewood Cliffs, N.J.: Prentice-Hall.

Rich, E.A. 1983. Users are individuals: Individualizing user models. International Journal of Man-Machine Studies, 18: 199-214.

Schank, R. and Hunter L. 1985. The quest to understand thinking. Byte, 10 (4): 143-155.

Shiffrin, R.M. and Dumais, S.T. 1981. The development of automatism. Cognitive skills and their acquisition. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Sleeman, D. 1977. A system which allows students to explore algorithms. Proceedings of the International Joint Conference on Artificial Intelligence, 780-786.

Sleeman, D. and Hendley, R.J. 1982. ACE: A system which analyses complex explanations. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Sleeman, D. 1984a. UMFE: A user modeling front end system. Heuristic Programming Project Working Paper HPP-84-12, Stanford University Department of Computer Science.

Sleeman, D. 1984b. Basic algebra revisited: A study with 14 year olds. Heuristic Programming Project HPP-83-9, Stanford University Department of Computer Science.

Sleeman, D. 1984c. Mis-generalization: An explanation of observed mal-rules. Proceedings of the Sixth Anmual Conference of the Cognitive Science Society.

Snow, R. E., Federico, P., Montague, W. E., eds. 1980. Aptitude, learning, and instruction, Vol. 2 of Cognitive process analyses of learning and problem solving. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Stevens, A. and Steinberg, C. 1981. A typology of explanations -- Application to ICAI. Report No. 4626, Bolt Beranek and Newman, Inc., Cambridge, Massachusetts.

Stevens, A. et al. 1981. STEAMER: Advanced computer aided instruction in propulsion engineering, Report No. 4702, Bolt Beranek and Newman, Inc., Cambridge, Massachusetts.

Stevens, A., Collins, A., and Goldin, S.E. 1982. Misconceptions in students' understanding. In Intelligent Tutoring Systems, ed. D. Sleeman, and J.S. Brown. London: Academic Press.

Stevens, A., and Collins, A. 1980. Multiple conceptual models of a complex system. In Aptitude, learning, and instruction. Vol. 2 of Cognitive process analyses of learning and problem solving. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Swartout, William R. 1983. XPLAIN: A System for creating and explaining expert consulting programs. Artificial Intelligence, 21: 285-325.

Tou, F.N. et al. 1982. RABBIT: An Intelligent Database Assistant. Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania, 314-318.

Tulving, E. 1983. Elements of episodic memory. New York: Oxford University Press.

van Dam, A. 1984. Computer software for graphics. Scientific American, 251 (3): 146-161.

VanLehn, K. 1981. Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. Xerox Palo Alto Research Center.

VanLehn, K. 1983. Human procedural skill acquisition: Theory, model, and psychological validation. Proceedings of the National Conference on Artificial Intelligence, Washington, D.C., 420-423.

VanLehn, K. and Brown, J.S. 1980. Planning nets: A representation for formalizing analogies and semantic models of procedural skills. In Aptitude, learning, and instruction, Vol. 2 of Cognitive process analyses of learning and problem solving, eds. R.E. Snow, P.Federico, and W.E. Montague, Hillsdale, N.J.: Lawrence Erlbaum Associates.

Wallis, J.W. and Shortliffe, E.H. 1982. Explanatory power for medical expert systems: Studies in the representation of causal relationships for clinical consultations. Methods of Information in Medicine, 21: 127-136.

Weiner, J. 1980. BLAH: A system which explains its reasoning. Artificial Intelligence, 15: 19-48.

Wickelgren, M.A. 1976. Memory storage dynamics. In Handbook of Learning and Cognitive Processes, Vol. 4. ed. W.K. Estes. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Winograd, T. 1977. A framework for understanding discourse. AIM-297. Stanford University Department of Computer Science.

Young, R. M. and O'Shea, T. 1981. Errors in children's subtraction. Cognitive Science, 5 (2): 153-177.